

(19)日本国特許庁 (JP)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開2001-111996

(P2001-111996A)

(43)公開日 平成13年4月20日(2001.4.20)

(51)Int.Cl.<sup>7</sup>

H 04 N 7/24

G 06 T 3/00

7/20

識別記号

5 0 0

2 0 0

F I

G 06 T 3/00

7/20

H 04 N 7/13

テーマコード<sup>\*</sup>(参考)

5 0 0 A

2 0 0 Z

Z

審査請求 未請求 請求項の数17 O L (全30頁) 最終頁に続く

(21)出願番号 特願2000-237435(P2000-237435)

(22)出願日 平成12年8月4日(2000.8.4)

(31)優先権主張番号 特願平11-221424

(32)優先日 平成11年8月4日(1999.8.4)

(33)優先権主張国 日本 (JP)

(71)出願人 000003078

株式会社東芝

神奈川県川崎市幸区堀川町72番地

(72)発明者 堀 修

神奈川県川崎市幸区小向東芝町1番地 株

式会社東芝研究開発センター内

(72)発明者 金子 敏充

神奈川県川崎市幸区小向東芝町1番地 株

式会社東芝研究開発センター内

(74)代理人 100058479

弁理士 鈴江 武彦 (外6名)

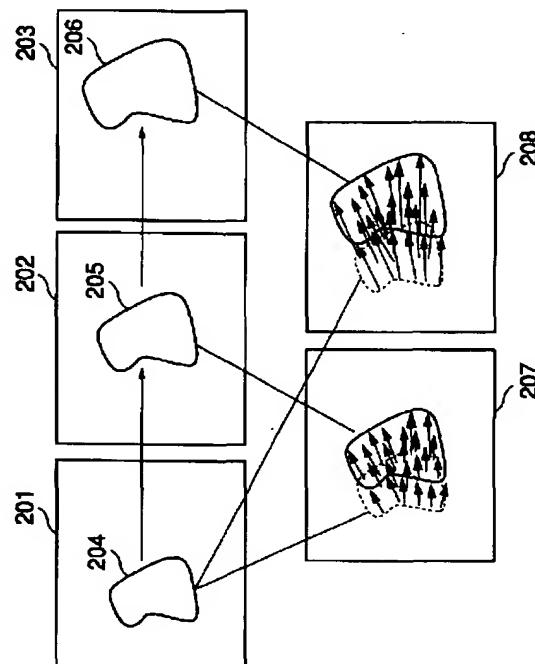
最終頁に続く

(54)【発明の名称】 物体領域情報記述方法及び物体領域情報生成装置並びに映像情報処理方法及び映像情報処理装置

(57)【要約】

【課題】映像データ中の所望の物体を少ないデータ量で記述でき且つその作成やそのデータの扱いも容易にする物体領域情報記述方法を提供すること。

【解決手段】映像データ(201～203)中における物体(204～205)を表す物体領域データを記述するための物体領域情報記述方法であって、対象物体領域(205、206)を、参照物体領域(204)から当該対象物体領域(205、206)への領域変換を示す変換パラメータを用いて表現し、変換パラメータの各々について、該変換パラメータをフレームの進行に沿って並べたときの軌跡を所定の近似関数で近似し、変換パラメータの各々についての、所定の近似関数を特定する近似関数パラメータと、該参照物体領域とを用いて、物体の領域に関する物体領域データを記述する。



1

## 【特許請求の範囲】

【請求項1】 複数フレームからなる映像データ中における物体の領域の時系列的な変化に関する物体領域データを記述する物体領域データ記述方法であって、前記物体の領域の中から少なくとも一つを参照物体領域として特定し、前記参照物体領域から対象となる物体領域への変換を示す変換パラメータを生成し、前記変換パラメータおよび前記参照物体領域に関する情報を用いて、前記物体領域データを記述する物体領域情報記述方法。

【請求項2】 複数フレームからなる映像データ中における物体の領域の時系列的な変化に関する物体領域データを記述する物体領域データ記述方法であって、前記物体の領域の中から少なくとも一つを参照物体領域として特定し、前記参照物体領域から対象となる物体領域への変換を示す変換パラメータを生成し、前記変換パラメータの時系列的な変化を所定の近似関数で近似し、前記所定の近似関数を特定する近似関数パラメータと、前記参照物体領域に関する情報を用いて、前記物体領域データを記述する物体領域情報記述方法。

【請求項3】 前記参照物体領域に関する情報はピットマップで表現されることを特徴とする請求項1または請求項2に記載の物体領域情報記述方法。

【請求項4】 複数フレームからなる映像データ中における物体の領域の時系列的な変化に関する物体領域データを記述する物体領域記述方法であって、前記物体の領域の中から少なくとも一つを参照物体領域として特定し、前記参照物体領域に対する近似図形の代表点から、対象となる物体領域に対する近似図形の代表点への変換パラメータを生成し、前記変換パラメータおよび前記参照物体領域に対する近似図形の代表点に関する情報を用いて前記物体領域データを記述する物体領域情報記述方法。

【請求項5】 複数フレームからなる映像データ中における物体の領域の時系列的な変化に関する物体領域データを記述する物体領域記述方法であって、前記物体の領域の中から少なくとも一つを参照物体領域として特定し、

前記参照物体領域に対する近似図形の代表点から、対象となる物体領域に対する近似図形の代表点への変換パラメータを生成し、前記変換パラメータの時系列的な変化を所定の関数で近似し、前記所定の近似関数を特定する近似関数パラメータおよび前記参照物体領域に対する近似図形の代表点に関する情報を用いて前記物体領域データを記述する物体領域

10

20

30

40

50

2

## 情報記述方法。

【請求項6】 前記変換パラメータを求めるにあたって、該変換パラメータに基づいて生成される予測物体領域と前記対象となる実際の物体領域との誤差が最小となるように、前記参照物体領域を更新することを特徴とする請求項1乃至請求項5のいずれか一項に記載の物体領域情報記述方法。

【請求項7】 前記参照物体領域として、前記物体の領域が存在する連続した複数フレームのうち中間に位置するフレームの当該物体の領域を選択することを特徴とする請求項1乃至請求項6のいずれか一項に記載の物体領域情報記述方法。

【請求項8】 前記参照物体領域をもとにして、該参照物体領域を含むフレームに後続するフレームにおける前記対象となる物体領域についての前記変換パラメータを求めるとともに、

該変換パラメータに基づいて生成される予測物体領域と前記対象となる実際の物体領域との誤差が予め定められたしきい値を上回った場合には、新たなフレームにおける前記対象となる物体領域を新たな参照物体領域として更新することを特徴とする請求項1乃至請求項7のいずれか一項に記載の物体領域情報記述方法。

【請求項9】 連続する複数フレームの中央に位置するフレームにおける前記物体の領域を前記参照物体領域とし、該参照物体領域をもとにして、該中央に位置するフレームの前後に連続するフレームにおける前記対象となる物体領域についての前記変換パラメータを求める処理を行い、

該変換パラメータに基づいて生成される予測物体領域と前記対象となる実際の物体領域との誤差が予め定められたしきい値を上回る範囲に属する連続する複数フレームに対して、前記処理を再帰的に行うことの特徴とする請求項1乃至請求項5のいずれか一項に記載の物体領域情報記述方法。

【請求項10】 前記対象となる物体領域を、複数の分割領域に分割し、この分割によって得られた前記分割領域の各々を独立して前記変換パラメータを生成することを特徴とする請求項1乃至請求項9のいずれか一項に記載の物体領域情報記述方法。

【請求項11】 前記物体に関連付けられている関連情報または該関連情報へのアクセス方法を示す情報を併せて記述することを特徴とする請求項1乃至請求項10のいずれか一項に記載の物体領域情報記述方法。

【請求項12】 複数フレームからなる映像データ中において物体の領域が存在する複数フレームの先頭のフレームおよび最終のフレームを特定可能な情報のデータ領域と、該複数フレームのうち任意のフレームにおける該物体の領域を特定する情報のデータ領域と、該任意のフレームにおける物体領域から他のフレームにおける物体

領域への変換に関する情報のデータ領域とを含む物体領域データを記録したコンピュータ読み取り可能な記録媒体。

【請求項13】複数フレームからなる映像データ中における物体の領域の時系列的な変化を所定の参照物体領域から対象となる物体領域への変換を示す変換パラメータを用いて記述した物体領域データを用いて前記映像データを表示している画面において特定の物体が指定されたか否かを判定するための映像情報処理方法であって、前記映像データを表示している画面において所定の任意の位置が指定された際、特定の物体に関する前記物体領域データから前記変換パラメータを取得し、前記変換パラメータを用いて、前記所定の位置を、前記参照物体領域が存在するフレームにおける位置に逆変換し、

この逆変換により求められた位置が、前記参照物体領域の内部に存在するか否かを調べ、内部に存在すると判断された場合に前記物体が指定されたか否かを判定することを特徴とする映像情報処理方法。

【請求項14】前記物体が指定されたと判定された場合には、前記物体に関連付けられている関連情報を呈示することを特徴とする請求項13に記載の映像情報処理方法。

【請求項15】複数フレームからなる映像データ中における物体の領域の時系列的な変化に関する物体領域データを記述する物体領域情報生成装置であって、前記物体の領域の中から少なくとも一つを参照物体領域として特定する処理部と、前記参照物体領域から対象となる物体領域への変換を示す変換パラメータを生成する処理部と、前記変換パラメータおよび前記参照物体領域に関する情報を用いて、前記物体領域データを記述する処理部とを具備する物体領域情報生成装置。

【請求項16】複数フレームからなる映像データ中における物体の領域の時系列的な変化に関する物体領域データを記述する物体領域情報生成装置であって、前記物体の領域の中から少なくとも一つを参照物体領域として特定する処理部と、前記参照物体領域から対象となる物体領域への変換を示す変換パラメータを生成する処理部と、前記変換パラメータの時系列的な変化を所定の近似関数で近似する処理部と、前記所定の近似関数を特定する近似関数パラメータと、前記参照物体領域に関する情報を用いて、前記物体領域データを記述する処理部とを具備する物体領域情報生成装置。

【請求項17】複数フレームからなる映像データ中における物体の領域の時系列的な変化を所定の参照物体領域から対象となる物体領域への変換を示す変換パラメータを用いて記述した物体領域データを用いて、前記映像

データを表示している画面において特定の物体が指定されたか否かを判定するための映像情報処理装置であつて、

前記映像データを表示している画面において所定の位置が指定された際、特定の物体に関する前記物体領域データから前記変換パラメータを取得する処理部と、

前記変換パラメータを用いて、前記所定の位置を、前記参照物体領域が存在するフレームにおける位置に逆変換する処理部と、

10 この逆変換により求められた位置が、前記参照物体領域の内部に存在するか否かを調べ、内部に存在すると判断された場合に前記物体が指定されたか否かを判定する処理部とを具備することを特徴とする映像情報処理装置。

#### 【発明の詳細な説明】

##### 【0001】

【発明の属する技術分野】本発明は、映像中の物体の領域に関する情報を記述するための物体領域情報記述方法、映像中の物体の領域に関する情報を生成するための物体領域情報生成装置、並びに映像中の物体に対する利用制御あるいは映像中の物体の検索を行う情報処理装置及びそのための映像情報処理方法に関する。

##### 【0002】

【従来の技術】映像中の物体領域を記述する方法として、従来はMPEG-4のオブジェクトコーディングが知られている。このMPEG-4のオブジェクトコーディングはフレーム毎にオブジェクト領域をビットマップとして表現し、さらに、フレーム内およびフレーム間で情報圧縮を行ない、形状およびテクスチャーは別々にコーディングされる。

30 【0003】しかし、このオブジェクトコーディングでは、任意形状でかつテクスチャーをもった物体領域をコンパクトに表現することを目的としているため、コーディングされたオブジェクトは、フレーム間圧縮されており、フレーム毎に物体領域の情報を取り出すことが困難であった。また、映像中の領域の位置情報を表現したい場合には、オーバースペックでありCodecに処理時間がかかるという問題があった。

##### 【0004】

【発明が解決しようとする課題】上述のようにMPEG-4のCodecを用いると、不必要的情報をコーディングするため、無駄が多いばかりでなく、フレーム間圧縮を行っているため、フレーム毎の情報を取り出すことが困難であるという問題があった。

【0005】本発明は上述した事情に対処すべくなされたもので、その目的は映像中の所望の物体の領域を少ないデータ量で記述でき且つその作成やそのデータの扱いも容易にする物体領域情報記述方法及び物体領域情報生成装置を提供することを目的とする。

50 【0006】また、本発明は、ユーザによる映像データ中の物体の指示の判定を容易にする物体領域情報記述方

## 5

法及び物体領域情報生成装置並びに映像情報処理方法及び情報処理装置を提供することを目的とする。

## 【0007】

【課題を解決するための手段】上記した課題を解決し目的を達成するために、本発明は以下に示す手段を用いている。

【0008】本発明は、映像データ中における任意の物体の領域に関する情報を連続する複数フレームに渡って記述するための物体領域情報記述方法であって、映像データ中における対象となる物体の領域を、所定の参照物体領域から該対象となる物体の領域への領域変換を示す変換パラメータで表現し、前記変換パラメータおよび前記参照物体領域に関する情報に基づいて、前記対象となる物体の領域に関する情報を記述することを特徴とする。

【0009】また、本発明は、映像データ中における任意の物体の領域に関する情報を連続する複数フレームに渡って記述するための物体領域情報記述方法であって、映像データ中における対象となる物体の領域を、所定の参照物体領域から該対象となる物体の領域への領域変換を示す変換パラメータで表現し、前記変換パラメータの各々について、該変換パラメータをフレームの進行に沿って並べたときの軌跡を所定の近似関数で近似し、前記所定の近似関数を特定する近似関数パラメータと、前記参照物体領域に関する情報を用いて、前記対象となる物体の領域に関する情報を記述することを特徴とする。

【0010】好ましくは、前記変換パラメータをそのまま記述することによって、前記対象となる物体の領域に関する情報を記述することを特徴とする。

【0011】好ましくは、前記参照物体領域を、ビットマップで表現することを特徴とする。

【0012】好ましくは、前記参照物体領域を、該物体領域に対する近似図形の代表点で表現することを特徴とする。

【0013】好ましくは、前記近似図形は、傾きのない矩形もしくは楕円、傾きのある矩形もしくは楕円、または多角形であることを特徴とする。

【0014】好ましくは、前記近似図形の種類に応じて、前記領域変換および前記変換パラメータの種類を選択することを特徴とする。

【0015】好ましくは、前記近似図形の代表点は予め順序付けられたものであることを特徴とする。

【0016】好ましくは、前記変換パラメータを求めるにあたって、該変換パラメータで生成される予測物体領域と前記対象となる物体の実際の領域との誤差が最小となるように、前記参照物体領域を選択することを特徴とする。

【0017】好ましくは、前記参照物体領域として、前記対象となる物体の領域が存在する連続した複数フレームにおける中間フレームの当該対象となる物体の領域を

## 6

選択することを特徴とする。

【0018】好ましくは、あるフレームにおける前記対象となる物体の領域を前記参照物体領域として、該参照物体領域をもとにして、該あるフレームに後続するフレームにおける前記対象となる物体の領域についての前記変換パラメータを求めるとともに、該変換パラメータで生成される予測物体領域と前記対象となる物体の実際の領域との誤差が予め定められたしきい値を上回った場合には、新たなフレームにおける前記対象となる物体の領域を新たな参照物体領域として更新することを特徴とする。

【0019】好ましくは、連続する複数フレームの中央に位置するフレームにおける前記対象となる物体の領域を前記参照物体領域とし、該参照物体領域をもとにして、該中央に位置するフレームの前後に連続するフレームにおける前記対象となる物体の領域についての前記変換パラメータを求める処理を行い、該変換パラメータで生成される予測物体領域と実際の物体の領域との誤差が予め定められたしきい値を上回る範囲に属する連続する複数フレームに対して、前記処理を再帰的に行うことを特徴とする。

【0020】好ましくは、対象となる物体の領域を、複数の領域に分割し、この分割によって得られた前記複数の領域の各々を独立して前記変換パラメータで表現することによって、前記対象となる物体の領域に関する情報を記述することを特徴とする。

【0021】好ましくは、前記物体に関する関連付けられている関連情報または該関連情報へのアクセス方法を示す情報を併せて記述することを特徴とする。

【0022】また、本発明は、映像データをもとにして上記した物体領域情報記述方法により記述された1または複数の物体の領域に関する情報を含む物体領域データを格納した記録媒体である。

【0023】また、本発明は、連続する複数のフレームからなる映像データを表示している画面において特定の物体が指定されたか否かを判定するための映像情報処理方法であって、映像データを表示している画面において任意の位置が指定された際、該映像データのフレーム中に存在する物体の領域に対応する、参照物体領域に関する情報を得て、該参照物体領域から該物体の領域への領域変換を示す変換パラメータの情報を取得し、前記変換パラメータを用いて、前記指定された位置を、前記参照物体領域が存在するフレームにおける位置に逆変換し、この逆変換により求められた位置が、前記参照物体領域の内部に存在するか否かを調べ、内部に存在すると判断された場合に前記物体が指定されたと判定することを特徴とする。

【0024】好ましくは、前記参照物体領域が代表点で表現されている場合には、該参照物体領域をビットマップに展開した後に、前記逆変換により求められた位置

が、該ビットマップに展開された参照物体領域の内部に存在するか否かを調べるようにもよい。

【0025】好ましくは、前記物体が指定されたと判定された場合には、前記物体に関連付けられている関連情報を表示するようにもよい。

【0026】また、本発明は、映像データ中における任意の物体の領域に関する情報を連続する複数フレームに渡って記述するための物体領域情報生成装置であつて、映像データ中における対象となる物体の領域を、所定の参照物体領域から該対象となる物体の領域への領域変換を示す変換パラメータで表現する手段と、前記変換パラメータおよび前記参照物体領域に関する情報に基づいて、前記物体の領域に関する情報を表現する手段とを備えたことを特徴とする。

【0027】また、本発明は、映像データ中における任意の物体の領域に関する情報を連続する複数フレームに渡って記述するための物体領域情報生成装置であつて、映像データ中における対象となる物体の領域を、所定の参照物体領域から該対象となる物体の領域への領域変換を示す変換パラメータで表現する手段と、前記変換パラメータの各々について、該変換パラメータをフレームの進行に沿って並べたときの軌跡を所定の近似関数で近似する手段とを備え、前記所定の近似関数を特定する近似関数パラメータと、前記参照物体領域に関する情報を用いて、前記対象となる物体の領域に関する情報を記述することを特徴とする。

【0028】また、本発明は、連続するフレームからなる映像データを表示している画面において特定の物体が指定されたか否かを判定するための情報処理装置であつて、映像データを表示している画面において任意の位置が指定された際、該映像データのフレーム中に存在する物体の領域に対応する、参照物体領域に関する情報および該参照物体領域から該物体の領域への領域変換を示す変換パラメータの情報を取得する手段と、前記変換パラメータを用いて、前記指定された位置を、前記参照物体領域が存在するフレームにおける位置に逆変換する手段と、この逆変換により求められた位置が、前記参照物体領域の内部に存在するか否かを調べ、内部に存在すると判断された場合に前記物体が指定されたと判定する手段とを備えたことを特徴とする。

【0029】なお、装置に係る本発明は方法に係る発明としても成立し、方法に係る本発明は装置に係る発明としても成立する。

【0030】また、装置または方法に係る本発明は、コンピュータに当該発明に相当する手順を実行させるための（あるいはコンピュータを当該発明に相当する手段として機能させるための、あるいはコンピュータに当該発明に相当する機能を実現させるための）プログラムを記録したコンピュータ読み取り可能な記録媒体としても成立する。

【0031】本発明では、例えば、連続または断続にフレーム中に存在する領域と表現したい物体の領域情報をフレーム毎に、近似図形またはビットマップで記述し、物体領域の存在する区間において、少なくともひとつの参照となる物体領域を選択する。その参照物体領域から他の物体領域への変換パラメータを計算し、変換パラメータを連続するフレーム順に並べて、変換パラメータの軌跡を描く。あるいは、さらに、この軌跡を、近似関数を用いて少ない情報量で表現を行う。この方法によつて、時空間情報を持った物体領域は、例えば、参照物体領域と変換パラメータの軌跡を近似した関数と物体領域の存在した区間の情報で表現される。また、例えば、任意のフレームの物体領域の情報を取り出すためには、近似関数から変換パラメータを求め、参照物体領域から、その変換パラメータを用いて物体領域を変形され予測物体領域を求め、その領域を求める物体領域とする。また、例えば、マウスポインタで指定された位置が時空間物体領域の中か外にあるか判定するには、求められた変換パラメータを用いて逆変換を行いマウスポインタの位置情報を参照物体領域のあるフレームの座標に変換し、参照物体領域の内か外かを判定し、指定されたフレームの物体領域の中にあるかを判定する。このように、容易に物体領域の内外判定ができるため、操作性の良いハイパーメディアアプリケーションを実現できる。

【0032】このように、本発明によれば、映像中を移動し、形が変化する物体領域の形および位置情報を少ない容量で記述することができ、フレーム毎の物体領域の情報を高速に取り出すことができる。特に、ハイパーメディアアプリケーションのように、マウスポインターを用いて、映像中の物体を指定した場合、マウスポインターの位置が物体の領域の中に含まれているか含まれていないかを高速に判定し、インタラクティブ性の高い操作環境を提供することができる。また、物体領域の動き、大きさなどを容易に知ることができ、検索システムや監視するシステムなどのアプリケーションに利用できる。

### 【0033】

【発明の実施の形態】以下、図面を参照して本発明による物体領域情報記述方法及び物体領域情報生成装置並びに映像情報処理方法及び情報処理装置の実施形態を説明する。

### 【0034】第1実施形態

図1に、本発明の第1の実施形態に係る物体領域情報生成装置の構成例を示す。図1に示されるように、本物体領域情報生成装置は、映像データ記憶部2、物体領域処理部4、変換パラメータ算出部6、関数近似部8、物体領域データ記憶部10を備えている。なお、本処理においてユーザの操作を介入させる形態を取る場合には、映像（動画像）データを例えばフレーム単位で表示させ、ユーザの指示入力等を受け付けるグラフィカルユーザインターフェース（GUI）が用いられる（図1では省略し

ている)。

【0035】物体(オブジェクト)領域は、映像中における一縦まりの領域部分であり、例えば、人、動物、植物、車、建物、道、川、太陽、雲など(あるいはその一部分、例えば人の頭、車のボンネット、建物の玄関など)、オブジェクトとして把握し得るものならどのようなものでも扱うことができる。

【0036】映像データ記憶部2は、映像データが記憶されているもので、例えばハードディスクや光ディスク、半導体メモリなどで構成される。なお、映像データ記憶部は他の処理部と同一サイトにある必要はなく、インターネット等を介して遠隔地に配置されていてもよい。

【0037】物体領域処理部4は、基準となるフレームの物体領域(参照物体領域)や対象となるフレームの物体領域(対象物体領域)を得るために処理を行う。

【0038】変換パラメータ算出部6は、参照物体領域をもとにして、対象物体領域の変換パラメータを算出する処理を行う。

【0039】関数近似部8は、物体領域の変換パラメータの各々について、その時系列的な軌跡を時間関数に近似する処理を行う。なお、後述するように、変換パラメータ自体を記述する場合には、この関数近似部8は不要となる。

【0040】物体領域データ記憶部10は、変換パラメータの各々について、その時系列的な軌跡を近似した関数式を表現するデータを含む物体領域データを記憶する。

【0041】なお、本実施形態では、参照物体領域の更新処理を行うことが好ましいが、その部分は図1では省略している。

【0042】映像データ記憶部2および物体領域データ記憶部10は、別々の記憶装置・記憶媒体によって構成されていてもよいが、それらの全部または一部が同一の記憶装置・記憶媒体によって構成されていてもよい。

【0043】また、本物体領域情報生成装置は、計算機上でソフトウェアを実行する形で実現することもできる。

【0044】図2に、本実施形態の物体領域情報生成装置の処理手順の一例を示す。

【0045】まず、ステップS101では、映像中の全てのフレームの物体領域(これは既知であるとする)を入力する処理を行う。例えば、GUIを介して入力する場合は、処理対象となる映像中の物体の輪郭をマウスやタッチパネル等のポインティングデバイスにより指定する。手動で入力された物体の輪郭内部を物体領域としてもよいし、また、入力された輪郭をSnakesとよばれる動的輪郭モデル(例えばM. Kass, A. Witkin and D. Terzopoulos著、「Snakes: Active contour model」、Proceedings of the 1st International Conference on Computer Vision, pp. 259-268, 1987年を参照)を用いた手

els」、Proceedings of the 1st International Conference on Computer Vision, pp. 259-268, 1987年を参照)を用いた手法により画像中の物体の輪郭線にフィッティングさせ、フィッティング後の輪郭の内部を物体領域としても良い。なお、手動で入力する代わりに画像処理により自動的に求めてもよいまた、物体領域に関するデータが既に存在している場合は、そのデータを読み込んでもよい。

【0046】この物体領域のうち、少なくとも一つを、参照物体領域として登録しておく。登録方法としては、物体の領域内には1を、領域外には0をそれぞれ対応させた2値のビットマップを作成し、記憶しておく方法がある。

【0047】また、参照物体領域の含まれるフレームを、参照フレームとして登録しておく。

【0048】次に、ステップS102では、参照物体領域から処理対象となっている1つのフレームにおける物体領域(以下、対象物体領域と呼ぶ)への変換パラメータを算出する処理を行う。

【0049】この処理は、例えば、対象物体領域内のオプティカルフロー算出処理とオプティカルフローからの変換パラメータ算出処理との2つの処理の組み合わせにより実現することができる。対象物体領域内のオプティカルフロー算出処理は、参照フレームから現フレームにかけての物体領域内の各画素(もしくは数画素からなるブロック)の動き(オプティカルフロー)を算出する処理である。

【0050】図3に、各フレームにおいて物体領域内のオプティカルフローを求める処理例の概要を示す。

【0051】図3において、201は参照フレーム、202は参照フレームの次のフレーム、そして203はフレーム202の次のフレームを表している。204, 205, 206はそれぞれのフレームにおける物体の領域である。207はフレーム201からフレーム202への物体領域のオプティカルフローである。また、208はフレーム201からフレーム203への物体領域のオプティカルフローである。

【0052】このようにオプティカルフローを求める方法は、参照物体領域と任意のフレームにおける物体領域を直接関係させることができるために、任意フレームの物体領域を算出したり、指定された座標が物体の内部か外部かを判定したりする処理が容易である。

【0053】図3の例では参照フレームから現フレームにかけての物体領域内の各画素(もしくは数画素からなるブロック)のオプティカルフローを求めたが、その代わりに、一つ前のフレームから現フレームにかけての物体領域内の各画素(もしくは数画素からなるブロック)のオプティカルフローを求めるようにしても良い。図4に、この場合の処理例の概要を示す。

11

【0054】図4において、301は参照フレーム、302は参照フレームの次のフレーム、そして303はフレーム302の次のフレームを表している。304、305、306はそれぞれのフレームにおける物体の領域である。307はフレーム301からフレーム302への物体領域のオプティカルフローである。また、308はフレーム302からフレーム303への物体領域のオプティカルフローである。

【0055】このようなオプティカルフローの算出方法は、図3の方法に比べてパラメータの変動が少なくなるが、任意フレームの物体領域の算出が面倒になる。図3と図4のどちらの方法を用いても構わないが、本実施形態では、図3の方法でオプティカルフローを算出するものとして説明を続ける。

【0056】なお、オプティカルフローを求める具体的な方法は多数提案されており（例えば、J. L. Barron, D. J. Fleet and S. S. Beauchemin著、「Performance of Optical Flow Techniques」、International Journal of Computer Vision, vol. 12, no. 1, pp. 43-77, 1994年を参

12

照）、どのような方法でも用いることが可能である。

【0057】また、参照物体領域内の特徴点を複数選択し、これら特徴点を中心とするブロックをテンプレートとするテンプレートマッチングにより求めた移動ベクトルをオプティカルフローの代わりに用いても良い。

【0058】次に、オプティカルフローから変換パラメータを算出する処理を行うが、求める変換パラメータは前提とする変換モデルにより異なる。

【0059】本実施形態では、パラメータ数が1のモデルとして、「拡大縮小モデル」、「回転モデル」、パラメータ数が2のモデルとして、「平行移動モデル」、パラメータ数が4のモデルとして、「拡大縮小／回転／平行移動モデルの複合モデル」（ここでは4-parameter変換モデルと呼ぶ）、パラメータ数が6のモデルとして、「affine変換モデル」、パラメータ数が8のモデルとして、「射影変換モデル」、パラメータ数が12のモデルとして、「parabolic変換モデル」が選択できるものとする。

【0060】これらのモデルを数式で表したのが数式(1)から数式(7)である。

【0061】

【数1】

13

$$\begin{cases} x' = a_0 x \\ y' = a_0 y \end{cases}$$

14

…(1)

$$\begin{cases} x' = x \cos a_0 - y \sin a_0 \\ y' = x \sin a_0 + y \cos a_0 \end{cases}$$

…(2)

$$\begin{cases} x' = x + a_0 \\ y' = y + a_1 \end{cases}$$

…(3)

$$\begin{cases} x' = a_0 x + a_1 y + a_2 \\ y' = a_1 x - a_0 y + a_3 \end{cases}$$

…(4)

$$\begin{cases} x' = a_0 x + a_1 y + a_2 \\ y' = a_3 x + a_4 y + a_5 \end{cases}$$

…(5)

$$\begin{cases} x' = \frac{a_0 x + a_1 y + a_2}{a_3 x + a_4 y + 1} \\ y' = \frac{a_5 x + a_6 y + a_7}{a_3 x + a_4 y + 1} \end{cases}$$

…(6)

$$\begin{cases} x' = a_0 x^2 + a_1 xy + a_2 y^2 + a_3 x + a_4 y + a_5 \\ y' = a_6 x^2 + a_7 xy + a_8 y^2 + a_9 x + a_{10} y + a_{11} \end{cases}$$

…(7)

【0062】数式(1)は拡大縮小モデル、数式(2)は回転モデル、数式(3)は平行移動モデル、数式(4)は4-parameter変換モデル、数式(5)はaffine変換モデル、数式(6)は射影変換モデル、数式(7)はparabolic変換モデルに相当する。数式中、参照物体領域中の座標を(x, y)、対象物体領域における物体の対応点の座標を(x', y')としている。各変換モデルでは、両フレームにおける対応点の関係が、それぞれの数式のようにa<sub>0</sub>~a<sub>11</sub>のパラメータを用いて表現できることを仮定している。もちろん、上述したモデル以外のパラメトリックなモデルを用意しても構わない。

【0063】変換パラメータの算出には、最小二乗法を用いることができる。これは、オプティカルフローにより求められた(x, y)と(x', y')との組み合わせを変換モデルの数式に代入した際に生じる誤差の2乗和を最小とするように変換パラメータを決定する方法である。これは古典的な方法で、行列演算で容易に実行することができる。

【0064】次に、ステップS103では、算出した物体領域の変換パラメータを時間関数へ変換(近似)する

30 処理を行う。

【0065】すなわち、ある時間区間におけるn個の変換パラメータa<sub>i</sub>(0 ≤ i ≤ n-1)(例えば、n=2)を、

a<sub>i</sub>=f<sub>i</sub>(t) (f<sub>i</sub>(t)は時間tの関数)

と表現する。この時間区間とは、同一の参照物体領域を用いてオプティカルフローを算出したフレームが含まれる時間区間である。また、f<sub>i</sub>(t)は多項式、スプライン関数、定数などとすることができる。

【0066】図5に、オプティカルフローから算出された(ある1つの)a<sub>i</sub>を関数により表現した様子を示す。図5において、401は関数表現の対象となるいる時間区間、402はオプティカルフローから算出されたa<sub>i</sub>の値、403はa<sub>i</sub>を表現する関数a<sub>i</sub>=f<sub>i</sub>(t)を表している。

【0067】このように関数により表現するメリットは、物体領域を記述するデータを少なくできることである。例えば、関数として2次以下の多項式を用いるとすると、この関数は3つの実数により表現することができるため、ある時間区間全てのパラメータの値を記述するのに3つの実数で十分になる。

15

【0068】パラメータを表現する関数を多項式やスプライン関数とする場合には、変換対象となっている時間区間における $a_i$ の値と $f_i(t)$ により算出される値との誤差が小さくなるように決定する。例えば、最小二乗法を利用すれば容易に計算ができる。

【0069】なお、この近似関数を求める処理は、例えば、当該物体領域に関する各フレームにおけるパラメータ値が得られる毎に行う（例えば各フレームのパラメータ値が得られる毎に近似を行うとともに近似誤差を求め、近似誤差が一定の範囲に収まるように近似区間を適宜分割する方法）ようにしてもよいし、参照物体領域が更新されて参照物体一定区間が定まった後に一括して行うようにしてもよい。

【0070】このステップS103の処理手順については後でより詳細に説明する。

【0071】次に、ステップS104では、参照物体領域の更新の必要性の判定処理を行う。

【0072】本実施形態は、任意のフレームにおける物体の領域を、参照フレームにおける参照物体領域と、参照物体領域の変換パラメータとにより表現するものである。しかし、表現したい物体領域と参照物体領域とがあまりに異なった形状である場合には、変換パラメータにより参照物体領域を移動／変形しても、表現したい物体領域に似ている形状が得られない。このような場合には、参照物体領域を別のフレームにおける物体領域に変更（参照領域の更新）するのが効果的である。そこで、本実施形態では、ステップS104にて、そのような変更が必要であるか否かを判定するようにしている。

【0073】この判定には、例えば、あるフレームの実際の物体領域と予測物体領域との誤差があらかじめ定められているしきい値を上回るかどうかにより判断する方法を用いることができる。予測物体領域とは、変換パラメータを用いて参照物体領域から算出したあるフレームの物体領域のことである。変換に用いる変換パラメータは、時間関数 $a_i = f_i(t)$ から算出した値を用いる。さらに、実際の物体領域と予測物体領域との誤差としては、例えば、両領域の共通部分の面積と共通しない部分の面積との比を用いることができる。

【0074】次に、ステップS105では、先のステップS104の判定処理において参照物体領域の更新が決定された場合に、参照物体領域更新処理を行う。この処理は、基本的にはステップS101の処理と同様である。すなわち、現在ステップS102の変換パラメータを算出する処理の対象となっているフレームを参照フレームとして登録し、参照物体領域を表現する2値のビットマップを作成する処理である。また、参照フレームにおける物体領域を参照物体領域として登録しておく。

【0075】次に、ステップS106では、映像中の物体領域の記述処理が終了かどうかを判定する。この判定基準としては、例えば、ビデオの終端かどうか、物体領

16

域存在区間の終端かどうか、ユーザにより終了が指示されたかどうか、などがある。ステップS102からステップS104またはステップS105までの処理が、ステップS106で終了と判定されるまでフレーム毎に繰り返し実行される。

【0076】そして、ステップS107において、これまでの処理により計算された物体領域の記述情報（変換パラメータを近似する関数のパラメータ）を、予め定められた記述フォーマットに従って記録する処理を行う。記録先は、例えば、計算機の内部もしくは外部の半導体メモリ、磁気テープ、磁気ディスク、光ディスクなどの物体領域データ記憶部10である。

【0077】図6に、本実施形態における1つの物体領域に対する物体領域データの記述フォーマットの一例を示す。

【0078】図6において、501は、物体IDで、物体に付与された物体固有の識別情報（例えば、番号または記号）である。502は、参照物体領域一定区間数で、同一の参照物体領域を有するフレーム区間の数である（図6ではNとしている）。これは参照フレームの数とも等しい。

【0079】503および504は、それぞれ、物体領域存在区間の先頭時刻および末尾時刻である。これらは時刻もしくはフレーム番号により記述される。末尾時刻の代わりに物体領域存在区間の長さ（時間もしくはフレーム番号の差分値）を用いることもできる。505は、物体領域記述情報である。物体領域記述情報505は、参照物体領域区間ごとに記述され、すなわち参照物体領域区間数（図6の例ではN個）だけ記述される。

【0080】各々の物体領域記述情報の具体的な内容は、図6中の506～510に示される。506および507は、それぞれ、参照物体領域区間の先頭時刻および末尾時刻である。これらは時刻もしくはフレーム番号により記述される。末尾時刻は参照物体領域区間の区間長で置き換えることができる。508は、変換モデルIDである。これは、拡大縮小モデル、affine変換モデル、parabolic変換モデルなどのうちどのモデルを用いて物体領域を記述したかを特定するためのIDである。511は、原点座標で、変換モデルの原点座標を画像のどこにとるかを決定するものである。原点座標データは、参照物体領域の重心位置を常に原点とするなどのルールを決めておく場合には省略することができる。509は、参照物体領域情報であり、参照物体領域を特定するための情報である。具体的には、参照フレームの時刻（またはフレーム番号）、参照物体領域を表すビットマップデータ（もしくはビットマップデータへのポインタ）である。なお、ビットマップデータは、そのままでデータサイズが大きいため、圧縮して保存するのが好ましい。510は、変換パラメータ情報である。変換パラメータ情報は、変換モデル（変換モデルID）

17

により定まるパラメータ数（図6の例ではM個）だけ記述される。変換パラメータは、具体的には、各フレームでのパラメータの値を示す配列や、パラメータを近似する関数を特定する情報（係数値など）などである。変換パラメータ情報の詳細については後で詳細に説明する。

【0081】以上の処理により、映像中の時空間的に変化する物体領域を簡単な記述データとして記録しておくことができる。

【0082】なお、本実施形態では、参照フレームを参照物体領域が一定の区間の先頭フレームとしたが、参照フレーム（および参照物体領域）は任意に選択することができる。他の参照物体領域選択方法については、後で（第5の実施形態にて）詳細に説明する。

【0083】次に本発明の他の実施形態を説明する。以下の実施形態で第1実施形態と対応する部分は同一参照数字を付して詳細な説明は省略する。

#### 【0084】第2実施形態

第2の実施形態は基本的には第1の実施形態と同様であり、以下では第1の実施形態と相違する点を中心に説明する。

【0085】本実施形態の物体領域情報生成装置の構成例は第1の実施形態（図1）と同様である。

【0086】図7に、本実施形態の物体領域情報生成装置の処理手順の一例を示す。

【0087】ステップS201では、ある物体領域が存在するフレーム区間の最初のフレームにおける物体領域のみをGUIを介して入力する。最初のフレームを参照フレームとして登録しておく。

【0088】ステップS202は、物体領域が存在するフレーム区間全体にわたり物体領域を計算する処理である。この具体的な手法としては、Snakesを利用することができる。

【0089】ステップS203は、図2のステップS102と同様に参照物体領域から処理対象となっているフレームの物体領域（対象物体領域）への変換パラメータを算出する処理である。この処理は、図2におけるステップS102のように、オプティカルフローを求める処理と、オプティカルフローから変換パラメータを推定する処理とにより実現できるが、本実施形態では物体領域の形状を用いて変換パラメータを算出する。具体的な方法としては、まず、対象物体領域を2値のビットマップにより表現する。次に、参照物体領域のビットマップの輪郭部分に隙間無くブロックを配置し、各ブロックごとにテンプレートマッチングにより対象物体領域のビットマップへの移動ベクトルを算出する。この移動ベクトルをオプティカルフローの代わりとして用いることにより、変換パラメータを算出する。

【0090】ステップS204～S208の処理は第1の実施形態で説明されているステップS103～S107の処理と同様である。

18

【0091】本実施形態における物体領域データの記述フォーマットの一例は第1の実施形態（図6）と同様である。

#### 【0092】第3実施形態

第1および第2の実施形態では、ビットマップで物体領域を表し、参照物体領域から処理対象となっているフレームの物体領域（対象物体領域）への変換パラメータを算出したが、第3の実施形態では、物体領域を近似图形で近似し、参照物体領域の近似图形の各代表点から処理対象となっているフレームの物体領域（対象物体領域）の近似图形の対応する各代表点への変換パラメータを算出するものである。本実施形態はこの相違点に対応する部分以外は基本的には第1、第2の実施形態と同様であり、以下では第1、第2の実施形態と相違する点を中心に説明する。

【0093】図8に、本実施形態に係る物体領域情報生成装置の構成例を示す。図8に示されるように、本物体領域情報生成装置は、映像データ記憶部2、物体領域処理部4、图形近似部5、変換パラメータ処理部6、関数近似部8、物体領域データ記憶部10を備えている。なお、本処理においてユーザーの操作を介入させる形態を取る場合には、映像（動画像）データを例えばフレーム単位で表示させ、ユーザーの指示入力等を受け付けるGUIが用いられる（図8では省略している）。

【0094】图形近似部5は、物体領域を近似图形で近似し、その近似图形の各代表点を求める処理を行うものである。

【0095】変換パラメータ算出部6は、基準となる参照フレームの参照物体領域の近似图形の代表点をもとにして、対象となるフレームの対象物体領域の近似图形の代表点への変換パラメータを算出する。

【0096】関数近似部8は、物体領域の近似图形の代表点の変換パラメータの各々について、その時系列的な軌跡を時間関数に近似する。なお、第1、第2の実施形態と同様に、変換パラメータ自体を記述する場合には、この関数近似部8は不要となる。

【0097】もちろん、本物体領域情報生成装置も、計算機上でソフトウェアを実行する形で実現することもできる。

【0098】図9に、本実施形態の物体領域情報生成装置の処理手順の一例を示す。

【0099】ステップS301は、図2におけるステップS101、あるいは図7におけるステップS201、S202と同様の処理である。ステップS302は、物体領域存在区間全体にわたり、物体領域をあらかじめ決められた图形により近似する処理である。

【0100】物体領域を图形近似する処理では、例えば、物体領域を囲むなるべく小さな近似图形を見つける。近似に用いる图形としては、傾きのないあるいは傾きのある矩形（正方形、長方形）、平行四辺形、傾きの

ないあるいは傾きのある楕円（円を含む）、多角形など、種々の図形を用いることができる。また、領域の近似方法も、領域に外接する図形に近似する方法、領域に内接する図形とする方法、領域の重心を近似図形の重心とする方法、領域と近似図形の面積を等しくする方法、領域と近似図形とが重ならない部分の面積を最小とする方法など、種々の方法がある。

【0101】なお、物体の領域を予め定められた图形により近似するのではなく、対象物体（オブジェクト）毎に图形の種類をユーザが指定できるようにしてもよいし、対象物体毎にその物体の形状等に応じて图形の種類を自動的に選択するようにしてもよい。

【0102】また、物体領域を近似する图形近似を求める処理は、フレームごとに行っても良いし、もしくは対象フレームの前後数フレームの物体領域を使って行っても良い。後者の場合には、近似图形の大きさや位置などの変化を数フレームの間で平滑化することにより、近似图形の動きや変形をなめらかにしたり、物体領域の抽出誤差を自立たなくすることができます。なお、近似图形の大きさは、フレームごとに異なって構わない。

【0103】物体領域の近似图形が求められたならば、この近似图形を表現する代表点を抽出する処理が行われる。どのような点を代表点とするかは、どのような近似图形を用いるかにより異なる。例えば、近似图形が矩形の場合には4つもしくは3つの頂点を代表点とすることが可能、近似图形が円の場合には中心と円周上的一点としたり直径の両端点としたりすることができる。また、楕円の場合には楕円の外接矩形の頂点としたり2つの焦点と楕円上の1点（例えば短軸上の1点）としたりすればよい。任意の閉多角形を近似图形とする場合には、各頂点を图形の代表点とする必要がある。

【0104】代表点の抽出は、1フレーム分の近似图形が得られるたびに、フレーム単位で行う。また、各代表点は、水平方向の座標xと、垂直方向の座標yと、により表される。

【0105】ここで、図10に、物体の領域が平行四辺形で表される場合に、近似楕円を求める方法の一例を示す。図10における点A, B, C, Dが物体の領域である平行四辺形の各頂点である。この場合、まず、辺ABと辺BCのどちらが長いかを計算する。そして、長い方の辺およびその対辺を辺の一部とする最小の長方形を求める。図10の例の場合は点A, B', C, D'を4頂点とする長方形となる。近似楕円は、例えば、この長方形に内接する楕円と相似で、かつ、点A, B', C, D'を通る外接楕円とする。

【0106】次に、楕円を表現する图形代表点を2つの焦点および1つの楕円上の点とした場合、楕円の焦点は2つの軸上の点や楕円の外接矩形から簡単に求めることができる。ここでは、一例として、図11における長軸上の2点P0, P1および短軸上の1点Hから焦点Fと

Gを求める方法を説明する。

【0107】まず、長軸と短軸のパラメータであるa, bと、楕円の中心Cと、扁平率eとを、

$$E(P0, P1) = 2 \times a,$$

$$C = (P0 + P1) / 2,$$

$$E(C, H) = b,$$

$$e = (1/a) \times \sqrt{(a \times a - b \times b)}$$

により求める。ここで、E(P, Q)は点Pと点Qのユークリッド距離である。

【0108】このようにして求めたパラメータから、焦点FとGは、

$$F = C + e \times (P0 - C),$$

$$G = C - e \times (P0 - C)$$

により求めることができる。

【0109】このようにして、楕円の代表点F, GおよびHは決定されるが、これらの点を別のフレームにおいて取り出された楕円の代表点と結びつける際にはあいまいさが生じる。すなわち、抽出された2つの焦点と1つ前のフレームにおける2つの焦点とを結びつける組み合

せは2通り存在する。また、短軸と楕円との交点は2つあるため、1つ前のフレームにおいて抽出された楕円上的一点と対応する交点がどちらなのかがわからない。そこで、これらを決定する方法について説明する。

【0110】1フレーム前に抽出された2つの焦点をFp, Gpとする。Fpに対応するのがFであるのかGであるのかを判断するために、E((Gp - Fp) / 2, (G - F) / 2)とE((Gp - Fp) / 2, (F - G) / 2)とを比較する。前者の方が小さい場合には、FpはFに対応させ、GpはGに対応させる。一方、後者の方が小さい場合には、その逆に、FpはGに対応させ、GpはFに対応させる。

【0111】また、1つ前のフレームにおける短軸と楕円との交点をHpとし、現フレームの短軸と楕円との2つの交点をH, H'とする。Hpと対応付ける点としてHとH'のどちらを選択するかは2つの距離、E(Hp - (Gp + Fp) / 2, H - (F + G) / 2)とE(Hp - (Gp + Fp) / 2, H' - (F + G) / 2)とを算出することにより決定する。前者が小さい場合にはHを選択し、そうでない場合にはH'を選択する。なお、はじめのフレームにおける短軸と楕円との交点Hは2つのうちのどちらを選択してもよい。

【0112】次に、近似图形として楕円以外の图形を用いた場合について説明する。

【0113】図12は、近似图形として平行四辺形を用いた場合の代表点を説明するための図である。点A, B, C, Dが平行四辺形の頂点である。これらのうち3点が決まれば残りの1点も一意に決まるため、代表点としては4頂点のうちの3頂点とすれば十分である。この例では、A, B, Cの3点を代表点としている。

【0114】また、図13は、近似图形として多角形を

用いた場合の代表点を説明するための図である。多角形の場合には、頂点の順序を外周に沿った順にしておく。図13の例では、10個の頂点を持つ多角形なので、N<sub>1</sub>～N<sub>10</sub>までの全ての頂点を代表点とする。なお、この場合に、内角が180度未満の頂点のみを代表点とするなどして、頂点数を削減するようにしてもよい。

【0115】さて、上記のように、近似図形は、図形を特定する代表点の集合により表現できる。図14の

(a)～(d)に、幾つかの種類の近似図形の代表点の例をまとめて示す。図14の(a)は矩形、図14の(b)は橢円、図14の(c)は平行四辺形、図14の(d)は多角形の各々について、黒丸で表されているのが代表点である。

【0116】ここで、近似図形の代表点は、位置関係が明らかになるように順序づけられているのが好ましい。図14の(a)～(d)の例においては、順序を番号で示しており、矩形および平行四辺形の代表点は、4頂点のうちの3頂点（時計回りに順序づけられる）としている。また、橢円は外接矩形の代表点（時計回りに順序づける）、多角形は全ての頂点（時計回りに順序づける）を図形の代表点とする。

【0117】また、内部に穴の空いた領域を表現することも可能である。図15に、その表現方法の一例を示す。図15では、二つの多角形により穴の空いた領域を表現している。図形の代表点は外側および内側の多角形の頂点である。このとき、外側の多角形の頂点は時計回りに順序づけており、内側の多角形の頂点は反時計回りに順序づけておく。このようにすることにより、常に代表点の進行方向の右側を物体の領域とすることができ、多角形の内部が物体なのか、外部が物体なのかの判定が容易に行えるようになる。

【0118】なお、外側の多角形の頂点を反時計回り、内側の多角形の頂点を時計回りに順序づけてももちろん構わない。この場合には、代表点の進行方向の左側が物体の領域となる。

【0119】統いて、このステップS302では、参照物体領域と参照フレームを設定する。本実施形態においては、参照物体領域は、物体領域存在区間の先頭のフレーム（参照フレーム）における物体領域近似図形である。また、参照物体領域近似図形の代表点の位置を併せて記憶しておく。

【0120】次に、ステップS303では、処理対象としているフレームの物体領域近似図形の代表点と、参照物体領域近似図形の代表点とを対応させる。

【0121】図16に、代表点の対応づけ方法の一例を示す。図16において、1000は近似矩形（複数）の重心である。図16では、参照物体領域近似図形1001と対象物体領域近似図形1002が得られている。

【0122】まず、近似図形1001と近似図形1002のいずれかを平行移動させ、重心位置を一致させる

（重心位置を一致させた状態が図16である）。次に、それぞれの矩形の頂点同士の距離d1～d4を計算し、全ての頂点の組み合わせで距離の和を求める。そして、距離の和が最小となる組み合わせを求め、対応づける。なお、この方法では対応付けが難しい場合がある。例えば、近似矩形が正方形に近い形状であり且つ45度回転移動している場合には、対応付けが難しい（2通りの組み合わせにおいて距離の和が同じような値になる）。

【0123】そこで、そのような場合には、近似図形内の物体領域同士の排他的論理和を取り、その面積が最小となる組み合わせを採用する方法、あるいは物体領域のテクスチャの絶対差分を求め、差分値が最小となる組み合わせを求める方法などを用いれば良い。

【0124】ステップS304では、物体領域近似図形の代表点の移動ベクトルから変換パラメータを算出する。

【0125】この処理では、代表点の動きをオプティカルフローの代わりに用いて、図2のステップS102と同等の処理により変換パラメータを算出する。ただし、代表点の数は少ないため、必ずしも変換パラメータが求まるとは限らない。例えば、矩形、橢円、平行四辺形は3つの代表点を持つが、この3つの移動ベクトルから射影変換モデルのパラメータ8個を求ることはできない。図17に、近似に用いる図形の種類と、変換パラメータを求めることができる変換モデルとの関係を示す。図17中の○はパラメータが算出できる組み合わせ、×は算出できない組み合わせである。

【0126】ステップS305は、ステップS304で求めた変換パラメータを時間関数により近似する処理であり、図2のステップS103と同じ処理である。

【0127】ステップS306は、参照物体領域を更新する必要があるか否かを判定する処理である。この処理では、まず、変換パラメータにより参照物体領域を変換し、現フレームの予測物体領域を算出する。もちろん、参照物体領域の代表点のみを変換パラメータにより変換し、変換後の代表点により特定される図形を構成しても同じ予測物体領域が算出できる。次に、予測物体領域と現フレームの対象物体領域近似図形との誤差を計算し、しきい値処理により参照物体領域更新を行うか否かを判定する。

【0128】ステップS307は、ステップS306において参照物体領域の更新が必要と判定された場合に、実際に更新を行う処理である。処理対象のフレームを参照フレームとし、当該フレームの物体領域近似図形を新たな参照物体領域として記憶し、参照物体領域の代表点の座標値も併せて保持しておく。

【0129】ステップS308は、図2のステップS106と同様に映像中の物体領域の記述が終了かどうかを判定する。

## 23

【0130】ステップS309は、図2のステップS107と同様に算出した物体領域の情報（変換パラメータを近似する関数のパラメータ）を予め定められた記述フォーマットで記録する処理である。

【0131】図18に、物体領域データの記述フォーマットの一例を示す。この記述フォーマットは、図形情報1109以外は、図6に例示した記述フォーマットと同じである。図6の参照物体領域情報509の代わりの図形ID1109は、図形の種類を特定するIDと、参照物体領域の近似図形の代表点の座標とからなる。図18中のMはIDにより特定される図形に必要な代表点の数を表す。

【0132】さて、以上の説明では物体領域の近似図形は特に限定しなかったが、以下では、物体領域の図形近似のバリエーションとして、物体領域の矩形による近似方法について説明する。

【0133】例えば、図19のようにフレーム2800の物体領域2801に外接する矩形2802は、物体領域のビットマップをラスタースキャンし、対象画素が物体領域内であるときに、X座標とY座標のそれぞれについて、それまで保存されている最小値より小さい場合には最小値を更新し、それまで保存されている最大値より大きい場合には最大値を更新する、という手順を、全ての画素について繰り返しチェックすることによって、X座標とY座標のそれぞれについて物体領域を示す画素位置の最小値および最大値を求めれば、矩形2802の4つの頂点座標を簡単に得ることができる。

【0134】このような方法は処理が簡易な点で優れているが、例えば図20のように細長い物体3001が画面3000に対して斜めの姿勢で存在しているときは、近似矩形3002内には非物体領域が特に多く含まれてしまう。また、細長い物体が回転すると矩形3002の大きさ、形状が変化する。これらは、物体指定する際の弊害となる場合がある。

【0135】そこで、矩形の大きさができるだけ小さくなり（近似矩形内の非物体領域ができるだけ少なくなり）、対象物体の姿勢も反映させることのできる近似方法の一例を示す。

【0136】図21の(a)において、3100は処理対象となっている映像中の1フレームを示している。

【0137】3101は抽出対象となっている物体の領域を示している。

【0138】3102は物体の領域を矩形で近似したものである。この近似矩形は図19の矩形2802とは異なり、傾きを持っている。矩形内の非物体領域も少なく、対象が回転してもその形状は一定である。

【0139】図22に、この場合の処理手順の一例を示す。この処理手順は、対象物体領域の慣性主軸を求め、これに基づいて近似図形を求めるようにしたものである。

## 24

【0140】図21の(b)において、3103は、対象物体領域の重心を示している。

【0141】3104は、対象物体領域の慣性主軸を示している。3105は、3104に垂直な直線である。

【0142】マスク画像を $f(x, y)$ とすると、 $f(x, y)$ は領域内では1で、領域外では0である。対象領域の慣性モーメントは、  
 $m_{ij} = \sum \sum x^i y^j f(x, y)$

で表せる。

【0143】まず、対象物体領域の慣性モーメント $m_{20}, m_{02}, m_{11}$ を求める（ステップS70～S72）。

【0144】ここで、原点（ここでは重心）を通る直線 $y = x \tan \theta$ についての $f(x, y)$ の慣性モーメントは、

$m_\theta = \iint (x \sin \theta - y \cos \theta)^2 f(x, y) dx dy$   
 で得られる。 $\theta$ を変化させたときに $m_\theta$ を最小にする角度を $\theta_0$ とする。一通りの角度しかないとき、直線 $y = x \tan \theta_0$ を慣性主軸と呼ぶ。 $\tan \theta_0$ は、2次方程式、

$$\tan^2 \theta + \{ (m_{20} - m_{02}) / m_{11} \} \tan \theta - 1 = 0$$

の解として求まる。これにより重心3103の周りで $\tan \theta_0$ を求めると、対象物体の慣性主軸が得られる（ステップS73）。

【0145】次に、慣性主軸に平行で物体領域に外接する直線と、慣性主軸に垂直で物体領域に外接する直線を求める（ステップS74）。図21の(b)において、直線3106, 3107は、慣性主軸3104に並行な直線であり、対象物体領域に外接する。直線3108, 3109は、直線3105に平行な直線であり、対象物体領域に外接する。矩形3102は、直線3106, 3107, 3108, 3109によって形成される（ステップS75）。

【0146】なお、対象物体が円の場合には慣性主軸が求まらないが、このような場合には例えば前述した物体領域のビットマップをラスタースキャンし、物体領域を示す画素位置の最小値および最大値を求める手順で近似矩形を求めればよい。

【0147】ところで、矩形より楕円で物体領域を表現する方が適当な場合もある。図23に、物体の領域が矩形で表される場合に、その矩形から近似楕円を求める方法の一例を示す。図24に、この場合の処理手順の一例を示す。

【0148】図23において、対象物体領域3300と外接矩形3301が得られているものとする。

【0149】まず、近似矩形の内接楕円および外接楕円を求める（ステップS80）。

【0150】図23において、楕円3302は矩形3301の内接楕円であり、楕円3303は矩形3301の

外接楕円である。

【0151】次に、内接楕円3302の大きさを少しづつ外接楕円3303に近づけていき（ステップS8  
1）、物体領域を全て包含する楕円3304を求め（ステップS82）、近似楕円とする。なお、繰り返し処理において、一回に内接楕円3302の大きさを拡大する単位は、予め定めておいてもよいし、内接楕円3302の大きさと外接楕円3303の大きさの差分に応じて決定してもよい。

【0152】また、上記とは逆に、外接楕円3303の大きさを内接楕円3302に近づけていってもよい。この場合には、外接楕円3303は最初から物体領域を全て包含しているので、例えば、繰り返し処理において、始めて物体領域に包含されない部分を生じた楕円の、一回前における楕円を、近似楕円とすればよい。

【0153】なお、上記では、物体領域を全て包含する楕円を求めたが、その代わりに、例えば、物体領域の領域と近似楕円の領域とが重複しない領域部分の面積を最小とすることを基準として、近似図形を求めるようにしてよい。

【0154】次に、近似矩形や近似楕円の代表点を求める。矩形の代表点は4つもしくは3つの頂点を代表点とすることができる、楕円の場合は楕円の外接矩形の頂点としたり、2つの焦点と楕円上的一点としたりすることができる。

#### 【0155】第4実施形態

以上、各フレームの対象物体領域を参照フレームの参照物体領域からの変換パラメータにより記述することにより映像中の所望の物体の領域を少ないデータ量で記述でき、かつその生成や扱いも容易な物体領域情報生成装置を説明したが、次に第1～第3の実施形態におけるパラメータ情報（図6の510、図18の1110）の具体的な内容について詳細に説明する。

【0156】このパラメータ情報には、参照物体領域に対する各フレームの物体領域の変換パラメータが直接、あるいは間接的に格納されている。前述したように、変換パラメータを時間関数で表現し、その関数を特定する情報を記述する方法や、変換パラメータの値を直接記述する方法などがあるので、これら表現方法それぞれについて変換パラメータの記述フォーマットを説明する。

【0157】まず、図25に、変換パラメータ情報のデータ構造の一例を示す。

【0158】フレーム数1200は、このパラメータ情報に格納されている物体領域のフレーム数を示す。

【0159】参照物体領域が一定の区間の各フレームの物体領域の変換パラメータは配列として第1フレームから順に格納される（1201）。

【0160】このような変換パラメータ情報は、1フレームにつき個数Mだけ存在する。例えば、*a f f i n e* 変換の場合は、M=6となる。

【0161】次に、図26に、変換パラメータを関数を用いて近似した場合の変換パラメータ情報のデータ構造の一例を示す。

【0162】変換パラメータの近似にはスプライン関数を用いるものとする。

【0163】節点フレーム番号1300は、スプライン関数の節点を表しており、この節点まで多項式の係数のデータ1302が有効であることを示している。多項式の係数データの数は、スプライン関数の最高次数により変化する（最高次数をKとすると、係数データの数はK+1となる）。そのため、多項式次数1301を参照する。多項式次数1301の後には、多項式次数+1個に相当する数の多項式係数データ1302が続く。

【0164】また、スプライン関数は節点間で別の多項式で表現されるため、節点の数に対応した数の多項式が必要になる。従って、節点フレーム番号、多項式の係数などを含むデータ1303は、複数繰り返し記述される。節点フレーム番号が参照物体領域一定区間の最終フレームと等しくなった場合には、それが最後の多項式係数データであることを意味しているので、変換パラメータデータが終わることが分かる。

【0165】次に、図27に、変換パラメータを関数を用いて近似した場合の物体領域記述情報のデータ構造の他の例を示す。ここでは、多項式の最高次数を2次として説明する。

【0166】前述した例（図6、図26）では、多項式スプライン関数の全ての係数を記述していたのに対して、ここでの記述方法では、スプライン関数の節点の座標と、スプライン関数の2次の係数に関連する値との組合せにより、記述する。この記述方法の利点は、節点が容易に取り出せるため、大まかな物体の軌跡が簡単にわかるという点である。

【0167】以下、この記述方法について詳細に説明する。

【0168】図27中、変換モデルID1400は、物体の動きの近似に用いた変換モデルの種類を特定する。例えば、*a f f i n e* 変換や射影変換を指定できる。また、原点座標1412は、変換モデルに従った変換を行う際に、画像のどこを原点とするかを指定するものである。

【0169】参照物体領域情報1401は、前述の例（図6の505）と同様に参照物体領域の形状を記述する情報が格納される。

【0170】節点数1402は、スプライン関数の節点の数を表す。各節点に対応するフレームは時間として表され、節点時刻1403に格納される。節点時刻は、節点数だけあるため、配列1404として記述しておく。

【0171】同様に、各節点の変換パラメータ1405もそれぞれパラメータが配列1406として記述される。この配列はパラメータの個数M (*a f f i n e* 変換

の場合はM=6)だけ存在する。

【0172】一次関数フラグ1407は、節点間のスプライン関数として一次関数だけが用いられているかどうかを表す。一部分でも2次以上の多項式を用いる場合には、このフラグはオフにしておく。このフラグを用いることにより、近似関数として一次関数のみに使われる場合に以下で説明する関数特定情報1408を一つも記述しなくて済むため、データ量を削減できるというメリットがある。なお、必ずしもこのフラグは必要ではない。

【0173】関数特定情報に含まれる関数ID1409と、関数パラメータ1410はそれぞれ多項式スプライン関数の次数と、その係数を特定するための情報を表す。図28に、それらの一例を示す。ここで、ta, tbは連続する節点の時刻、fa, fbは時刻ta, tbにおける節点の変換パラメータを表している。1次多項式を用いるときは節点のみの情報で十分なので、関数パラメータは記述されないが、2次多項式の場合には係数を特定するための情報として一つのaaが関数パラメータに記述される。なお、図28の例では、2次の係数が用いられているが、例えば、二次曲線上のfa, fb以外の1点など、他の値を用いることもできる。

【0174】本実施形態の記述方法では、節点の情報と関数パラメータの情報により、図28の制約条件を用いて全ての区間ににおけるスプライン関数が再現できる。

【0175】関数特定情報は、(節点数-1)個存在し、これらは配列1411となって記述される。

【0176】なお、上記では、多項式の最高次数を2次として説明したが、もちろん、多項式の最高次数を3次以上とすることも可能である。

【0177】ここで、データ記述の具体例を用いて図27のデータ構造を用いた記述の効果を説明する。図29は図25で説明されているデータ構造で代表点軌跡データを記述した例(0000)と、図27のデータ構造を用いて代表点軌跡データを記述した例(0003)とを比較したものである。ただし、例(0000)ではパラメータ1201に加え、わかりやすいようにタイムスタンプも図示している。両者とも、式(7)による平行移動モデルを用いた場合の例であり、二つの変換パラメータa0, a1の軌跡の記述を図示している。例(0000)では、時刻0.0(秒)から時刻15.0(秒)までのパラメータa0の値が0.1(秒)おきに記述されている。これらの値は、既に説明した変換パラメータ処理部6、またはステップS102、またはS203、またはS304による処理で算出されたものである。パラメータ値にはa0, a1との2つがあるため、パラメータ値データ配列0000では $2 \times 15 = 30$ の座標値がメモリに格納されている。それぞれの値を4バイトのデータで表現する場合には、パラメータ値データ配列0000の格納には1208バイトのメモリ容量を必

要とする。

【0178】図27のデータ構造により物体領域を記述するには、これらのパラメータ値のデータを関数近似処理部8、またはステップS103、S204、またはS305により関数近似する。ここで、これらの処理のより具体的な例を図30の流れ図を用いて説明する。

【0179】ステップS0000は初期化処理であり、処理中に注目する区間の開始フレームのタイムスタンプtsおよび終端フレームのタイムスタンプteに初期値を代入する。ステップS0001は近似関数を算出する処理であり、tsからteまでのパラメータ値aiをパラメータ値データ配列(図29の0000)から取り出し、最小二乗法によりあらかじめ決められた時間tの関数fi(t)(この例では2次または1次の多項式)を決定する。この例ではi=0, 1である。ステップS0002では、tsからteの区間において、ステップS0001で算出された関数により算出されるパラメータ値の最大誤差を変数eにセットする。この際の最大誤差は、物体領域の誤差面積である。すなわち、実際の物体領域と、fi(t)から算出されるパラメータ値により参照物体領域を変換した領域との不一致領域の面積をtにおける誤差とし、tsからteまでの最大誤差にeをセットする。物体領域の誤差面積の代わりに、実際の物体領域に対する不一致領域の面積の比を誤差として用いてよい。ステップS0003の選択処理では、eの値とあらかじめ決められているしきい値とを比較し、eのほうが大きければステップS0004を、そうでなければステップS0005を次に行う。

【0180】ステップS0004では、時刻tsのフレームから時刻teのフレームの直前のフレームまでの区間に對して最小二乗法で算出された近似関数を登録し、tsに時刻teのフレームの直前のフレームにつけられているタイムスタンプをセットする。このステップにより、例えば図29における0001の関数 $a_0 = 2.1 t^2 - 12.6 t$ ( $0 \leq t \leq 2$ )が登録される。 $0 \leq t \leq 2$ は関数の有効時間区間であり、近似関数が算出された際のtsからteの区間がこれに相当する。この時間区間も関数と併せて登録される。

【0181】ステップS0005はteの更新処理であり、時刻teのフレームの次フレームに付与されたタイムスタンプにteの値を更新する。ステップS0005は終了判定処理であり、teが処理すべき範囲を超えているかどうかを判定し、超えていなければステップS0006からの処理を繰り返し、超えていればステップS0007の最終処理に進む。ステップS0007は最後の近似関数を登録する処理である。例えば、図29における0001の最後の関数 $a_0 = 0.2 t^2 - 4.4 t - 12.4$ ( $10.5 \leq t \leq 15$ )はこのステップで登録された関数である。

【0182】以上の処理により近似関数の算出が終わる

と、所定の記述フォーマットに従ってこれらの近似関数データが記録される。図29の0003は図27の記述フォーマットを用いている（図示しているのは図27の節点時刻配列1404、節点パラメータ1の配列1406、節点パラメータ2の配列、…、一次関数フラグ1(1407)、関数特定情報1の配列1411、一次関数フラグ2のそれぞれ一部である）。例えば、0001の最初の関数  $a_0 = 2.1 t^2 - 12.6 t$  ( $0 \leq t \leq 2$ ) は、両端の節点時刻として0.0と2.0が記録される。また、これらの時刻における  $a_0$  のパラメータ値である0.0と-16.8が記録される。さらに、2次多項式であることを表す関数IDとして2が、また、t<sup>2</sup>の係数2.1を特定するための関数パラメータとして4.2がそれぞれ記録される。関数パラメータとt<sup>2</sup>の係数との関係は図28の通りである。この記述フォーマットでは、1次多項式の場合には多項式の有効時間区間の両端の座標値だけが記録され、2次多項式の場合には2次係数の2倍の値も併せて記録される。

【0183】0003では、実数に4バイト、一次関数フラグに1ビット、関数IDに2ビットを費やす場合には、わずか57バイトのデータ容量しか必要としない。これは先に計算した0000に必要となるデータ容量1208バイトよりもかなり少ない値である。

【0184】このようなデータ容量の削減が可能である理由は、自然界の物体の動きは通常滑らかであるため、変換パラメータの変化も滑らかであるという特徴を利用しているからである。すなわち、物体の変形・移動を示すパラメータ（変換パラメータ）のデータ列は少々の誤差を許容することにより多項式でまとめて表現することが可能であり、この多項式を特定するパラメータだけを記録することで全ての変換パラメータ値を記録するよりも大幅にメモリ容量を節約できる。

【0185】なお、物体領域全体としては図29の0003に記述されている以外に、図27にある変換モデルID、原点座標、参照物体領域情報、節点数などのデータが必要である。以下では、物体領域データのデータ構造に関連するバリエーションについて説明する。

【0186】まず、以上では、ある1つの物体領域に関して全てのフレームを対象として変換パラメータを求めたが、変換パラメータを求めるフレームをサンプリングするようにしてもよい。例えば、3フレームに1フレームだけサンプリングし、フレーム1から参照物体領域、フレーム4, 7, …からそれぞれ対象物体領域を用いるなどである。

【0187】なお、変換パラメータを時間関数で表現し、その関数を特定する情報を物体領域データに記述する場合には、そのサンプリングされたパラメータ値によってこれまでと同様に関数近似を行えばよい。また、物体領域データにサンプリングに関する情報を含める必要はない。

【0188】一方、変換パラメータの値を物体領域データに直接記述する場合には、例えば、第1の実施形態において、（1）そのサンプリングされなかったフレームのパラメータ値を適宜補間し（例えば、直前にサンプリングされたフレームと同じ値を記述する）、物体領域データは図5と同様とするか、あるいは、（2）図31のように、物体領域データにサンプリング情報520を持たせ、サンプリングした場合にはそのサンプリングしたフレームのパラメータ値のみを記述し、サンプリング情報520にサンプリング方法を特定可能な情報（例えば、nフレームに1回サンプリングしたことを示す情報として数値n（ただし、例えばn=1の場合には全てのフレームからサンプリングされたことを示すものとする））を記述すればよい。なお、（2）の方法においては、この物体領域情報を使用する際にサンプリング情報520を参照することにより、サンプリングされなかったフレームのパラメータ値を必要に応じて補間することができる。以上の点は、他の実施形態の場合も同様である。

【0189】次に、以上の構成において、参照物体領域の更新判定処理を行わずに、参照物体領域を固定する構成も可能である。例えば、物体領域の動きが少ない場合や、あるいは、ある程度誤差が大きくても良いような場合などには、処理が簡易になるので有効である。この場合、物体領域データのデータ構造はこれまでのものと同じものを使用することができる（参照物体領域一定区間数が1となり、物体領域記述情報が1個となる）。また、常に参照物体領域の更新判定処理を行わない場合には、物体領域データのデータ構造はこれまでのものから参照物体領域一定区間数のフィールドを削除し、物体領域記述情報のフィールドを1個とするようにしてもよい。

#### 【0190】第5実施形態

統いて、これまでの各実施形態における参照物体領域選択方法の具体例について説明する。

【0191】図32は、予測物体領域と真の物体領域との誤差が最小となるように、物体領域存在区間から参照物体領域を選択する手法の一例を表したフローチャートである。

【0192】ステップS400では、物体領域存在区間から任意の1フレーム分の物体領域データを参照物体領域として取り出す。

【0193】ステップS401では、ステップS400の参照物体領域から物体領域存在区間の他フレームの物体領域への変換パラメータを求める。

【0194】ステップS402では、ステップS400の参照物体領域とステップS403で求めた変換パラメータを用いて、物体領域存在区間の参照フレームを除く全フレームの予測物体領域を求める。

【0195】ステップS403では、物体領域存在区間

の参照フレームを除く全フレームについて、ステップS 402で求めた予測物体領域と実際の物体領域との誤差を求め、それらを加算して誤差の合計値を求める。

【0196】ステップS 404では、ステップS 403で求めた誤差の合計値が、他のフレームを参照フレームとして以前に求めた誤差の合計値のいずれよりも小さい場合は、現在の参照フレーム番号を保存する。

【0197】ステップS 405は、分岐処理であり、現在の参照フレームが物体領域存在区間の最終フレームである場合にはステップS 406へ、そうでない場合にはステップS 400へと分岐する。ステップS 400へ分岐した場合には、新たなフレームを参照フレームとして、同様の処理を繰り返す。

【0198】ステップS 406では、ステップS 404で保存してあった予測物体領域と実際の物体領域との誤差の合計値が最小となる参照フレームの番号を取り出して、このフレームの物体領域を参照物体領域とする。

【0199】図32を用いた実施形態は計算量が大きいため、より高速な処理が求められる場合には、精度を犠牲にして、特定フレームを参照フレームとして選択すればよい。この特定フレームは、物体領域存在区間の先頭のフレームであってもよいが、中央のフレームを参照フレームとして用いれば、参照フレームと対象フレームの間隔のうち、最も離れた部分が最小となるため、多くの場合にはより精度のよい近似が期待できる。

【0200】図33は、物体領域存在区間の中央のフレームを参照フレームとして、参照物体領域を選択する手法の一例を説明する図である。

【0201】物体領域存在区間1700の先頭フレーム1701と最終フレーム1702の中間の中央フレーム1703を参照フレームとして選択して、他のフレームの物体領域の変換パラメータを求めている。

【0202】図34は、物体領域存在区間から、複数の参照フレームをリアルタイムに選択する手法の一例を表したフローチャートである。また、図35は、同手法を説明する図である。リアルタイム処理の場合、最終的な物体領域存在区間が確定していない場合、参照フレームを決定することが必要である。

【0203】ステップS 500では、物体領域存在区間から1フレーム分の物体領域データを参照物体領域として取り出す。図35中では先頭フレーム1900の物体領域データを取り出す。

【0204】ステップS 501は、分岐処理であり、現在の処理フレームが物体領域存在区間の最終フレームである場合には終了へ、そうでない場合にはステップS 502へと分岐する。

【0205】ステップS 502では、物体領域存在区間からステップS 500で取り出したフレームの次のフレームの物体領域データを対象物体領域として取り出し、ステップS 501の参照物体領域に対する変換パラメー

タを求める。

【0206】ステップS 503では、ステップS 501の参照物体領域とステップS 502で求めた変換パラメータを用いて予測物体領域を求める。

【0207】ステップS 504は、分岐処理であり、ステップS 503で求めた予測物体領域とステップS 502の実際の対象物体領域との誤差が閾値を越えているかを判定する。誤差が閾値を越えている場合はステップS 505へ、越えていない場合はステップS 501へ分岐する。

【0208】ステップS 501へ分岐した場合、現在のフレームが物体領域存在区間の最終フレームでなければ同様な処理を繰り返す。これらの同じ参照フレームに対して、繰り返し処理されるフレームの集合は図35中では1901に相当する。

【0209】ステップS 505では、現在のフレームを新たな参照フレームとし、ステップS 501へ進んで、現在のフレームが物体領域存在区間の最終フレームでなければ同様な処理を繰り返す。図35中では新たな参照フレーム1902に相当する。

【0210】参照フレームが更新されると、参照物体領域一定区間1903が確定する。物体領域存在区間は複数の参照物体領域一定区間にによって構成される。

【0211】図36は、物体領域存在区間から、複数の参照フレームを選択する別の手法を表したフローチャートである。また、図37の(a)、(b)は、同手法を説明する図である。

【0212】ステップS 600では、未処理の物体領域存在区間から中央のフレームの物体領域データを参照物体領域として取り出す。図37の(a)中では物体領域存在区間2100の中央フレーム2101が参照フレームとなる。

【0213】ステップS 601は、分岐処理であり、現在処理しているフレームが未処理の物体領域存在区間の先頭フレームかどうかを判定する。先頭フレームでない場合にはステップS 602へ、先頭フレームである場合にはステップS 606へと分岐する。

【0214】ステップS 602では、現在の処理フレームの1つ前のフレームの物体領域を取り出し、ステップS 600の参照物体領域に対する変換パラメータを求める。

【0215】ステップS 603では、ステップS 600の参照物体領域とステップS 602で求めた変換パラメータを用いて、予測物体領域を求める。

【0216】ステップS 604は、分岐処理であり、ステップS 603で求めた予測物体領域とステップS 602の実際の対象物体領域の誤差が閾値を越えているかを判定する。誤差が閾値を越えている場合はステップS 605へ、越えていない場合はステップS 601へ分岐する。ステップS 601へ分岐した場合、現在のフレーム

が未処理の物体領域存在区間の先頭フレームでなければ同様な処理を繰り返す。

【0217】ステップS605では、現在処理しているフレームの1つ後のフレームを参照物体領域存在区間の先頭フレームとする。図37の(a)、(b)中では、参照フレーム2101の物体領域を参照物体領域としたときに、フレーム2102(第kフレームとする)の物体領域で誤差が閾値を超えた場合、第k+1フレームが参照物体領域存在区間の先頭フレームとなる。

【0218】ステップS606では、未処理の物体領域存在区間から中央のフレームの物体領域データを参照物体領域として取り出す。

【0219】ステップS607～S611は前述のステップS601～605と同様の処理を、後のフレームに向かって行う。図37の(a)、(b)中で、フレーム2103の直前のフレームが参照物体領域一定区間の最終フレームになったとすると、未処理の物体領域存在区間2105と2106が残る。

【0220】ステップS612は、分岐処理であり、未処理の物体領域存在区間が存在していればステップS600へ、存在していない場合は処理を終了する。ステップS600へ分岐した場合、未処理の物体領域存在区間の中央フレームを新たな参照フレームとして、未処理の物体領域存在区間がなくなるまで、処理を繰り返す。図37の(b)中では、未処理の物体領域存在区間2105の中央フレーム2107を新たな参照フレームとして処理し、新たな参照物体領域一定区間2108と未処理の物体領域存在区間2109が生じた例を示している。

#### 【0221】第6実施形態

続いて、これまでの各実施形態において、1つの物体(オブジェクト)を複数の領域に分けて物体領域データを作成する方法について説明する。

【0222】これまでには、1つの物体に対して1つの変換パラメータを求めていた。しかし、見かけの形状が大きく変化するような物体の場合、物体をいくつかの領域に分割し、それぞれの領域に対して変換パラメータを用いる方が良い場合がある。例えば、歩行している人間では、頭や胴体の部分は動きが少ないが、手足は激しく動作する。人間を1つの物体として扱うよりも、頭/胴体/手/足と別の領域に分割する方が、各部の変換パラメータを安定して求めることができる。

【0223】1つの物体を複数の图形で表わす場合には、物体を複数の領域に分割する処理が必要となる。この処理にはどのような方法を用いてもよいが、例えば人手で直接入力する方法がある。この場合、例えば、マウス等のポインティングデバイスを用いて、画像上で領域を矩形や楕円で囲む、あるいはポインティングデバイスの軌跡により領域を指定する、などの操作で実現することができる。また、人手ではなく自動で行う場合には、例えば、物体の動きのクラスタリングで実現する方法が

ある。これは、連続するフレーム間で物体中の各領域がどのような動きをしたかを相関法(例えば、画像解析ハンドブック、第11部、第3章、東京大学出版会、1991を参照)や勾配法(例えば、Determining optical flow, B. K. P. Horn and B. G. Schunck, Artificial Intelligence, Vol. 17, pp. 185-203, 1981を参照)などにより求め、これらの動きの似ているものだけをまとめて領域を形成する方法である。

【0224】図38に、オプティカルフローの似ているものをまとめて領域分割を行う様子を示す。

【0225】また、図39に、物体を複数の領域で記述するためのデータ構造の一例を示す。これは物体を単一の領域で記述するデータ構造(図18)を拡張したものであり、2906以下のデータは図18と共通である。2902に分割した領域の数を保持し、2905以降に各領域のデータを保持する。さらに、図15に示すように穴の空いた領域についても、穴を1つの領域として表現し、図39のデータ構造で扱うことが可能である。

#### 【0226】第7実施形態

次に、これまでの各実施形態により得られる物体領域データの利用例について説明する。

【0227】例えば、物体領域データに該物体(オブジェクト)に関する関連情報を付加するようにもよい。この場合、映像を閲覧中のユーザが映像中の物体を指示する(例えばマウスでクリックする)ことにより、該物体の関連情報を表示するような、ハイパーテディアアプリケーションを実現することができる。

【0228】その際、映像や物体はどのようなものであってもよい。例えば、映像が映画などのコンテンツ、物体が俳優等の登場人物もしくは他の登場物体、関連情報がその俳優もしくはその役柄等に関する説明であれば、映画を視聴している視聴者は、所望の俳優に関する説明をその俳優の画像をクリックするだけで閲覧することができる。同様に、電子百科事典、電子カタログ等のあらゆる電子コンテンツに適用可能である。

【0229】関連情報は、文字、音声、静止画、動画、あるいはそれらを適宜組み合わせたものであってもよいし、プログラムもしくは計算機の動作を記述したデータであってもよい。前者の場合には情報が表示され、後者の場合にはプログラム等が実行される。

【0230】物体領域データには、関連情報を付加する代わりに、その関連情報を取得するためのアドレス等のポインタ情報を付加してもよい。

【0231】また、例えば、関連情報にキーワードを記述しておけば、物体をキーワード検索することができる。さらに、関連情報にその物体から抽出した、シェイプ、テクスチャ、アクティビティ、カラーなどの特徴量を記述しておけば、そのような特徴量をもとにオブジェ

クト検索することができる。

【0232】また、例えば、物体領域データを解析することにより得られる、物体のシェーブ、テクスチャ、アクティビティ、カラーなどの特徴量に基づいて、不審な人物等の監視を行う、監視システムを実現することができる。

【0233】次に、映像データや物体領域データの提供方法について説明する。

【0234】本実施形態の処理により作成された物体領域データがユーザの用に供される場合には、作成者側からユーザ側に何らかの方法で物体領域データを提供する必要がある。この提供の方法としても以下に例示するように種々の形態が考えられる。

(1) 映像データとその物体領域データとその関連情報を1つ(または複数の)記録媒体に記録して同時に提供する形態

(2) 映像データとその物体領域データとを1つ(または複数の)記録媒体に記録して同時に提供するが、関連情報は別途提供するかもしくは提供しない(後者は例えば提供しなくてもユーザがネットワーク経由等で別途取得できる場合)形態

(3) 映像データを単独で提供し、別途、物体領域データと関連情報を1つ(または複数の)記録媒体に記録して同時に提供する形態

(4) 映像データ、物体領域データ、関連情報を別々に提供する形態

上記は主に記録媒体により提供する場合であるが、その他にも、一部または全部を通信媒体で提供する形態も考えられる。

#### 【0235】第8実施形態

以下では、本発明の物体領域表現方法をハイパーテディアに応用した例について説明する。

【0236】物体(オブジェクト)に関連情報を結び付けるハイパーリンクでは、物体がユーザによって指定されたことを検出する必要がある。ここでは指定された点が時空間物体領域の内部にあるかあるいは外部にあるかを判定する方法の一例を示す。

【0237】なお、本実施形態では、ユーザが(主にGUI画面上で)物体を指示することにより、関連情報を表示(文字、静止画、動画等の表示、音声の出力等)させ、あるいは関連するプログラムを実行させるような場合を例にとって説明する。

【0238】図40に、本実施形態に係る情報処理装置の構成例を示す。図40に示されるように、本情報処理装置は、映像データ表示部301、制御部302、関連情報表示部303、指示入力部304を備えている。

【0239】映像データ表示部301は、図示しない記録媒体等から入力した映像データを液晶表示装置もしくはCRT等に表示するためのものである。

【0240】指示入力部304は、ユーザがマウス等の

ポインティングデバイスもしくはキーボードなどを用いて、液晶表示装置もしくはCRT等に表示された映像中の物体を指示するなどの操作を行うことを可能とし、そのユーザからの入力を受け付けるためのものである。

【0241】制御部302は、詳しくは後述するが、例えばユーザが画面上で指示した座標と図示しない記録媒体等から入力した物体領域データとに基づいてユーザが映像中の物体を指示したか否か判定する。

【0242】関連情報表示部303は、制御部302により物体が指示されたと判定されたときに、物体領域データに関連情報が付加されている場合には該関連情報を表示し、物体領域データに関連情報へのポインタ情報が付加されている場合には該ポインタ情報をもとに関連情報を(記録媒体やネットワークを介したサーバ等から)取得して表示する。

【0243】なお、関連情報がプログラムや計算機の動作を記述したデータである場合には、当該プログラム等が実行される。

【0244】また、物体が指示された場合に、関連情報の表示等を行うのではなく、他のアクションが取られる場合には、関連情報表示部303の代わりに該当する処理部が実装される。

【0245】もちろん、本実施形態もソフトウェアによっても実現可能である。

【0246】さて、以下、図41の例を用いて、本実施形態の処理手順について説明する。

【0247】図41において、参照物体領域を2201、対象フレームを2202とする。ユーザによって指定された点が2203であり、その点の参照フレームにおける位置が2204である。

【0248】図42に、本実施形態の処理手順の一例を示す。ただし、図42のフローチャートでは、映像の再生中に表示されている映像内をマウスカーソル等のポインティングデバイスを用いて指示された際に指定された点が時空間物体領域の内部にあるかあるいは外部にあるかを判定する処理についてのみ示している(基本的には制御部302の処理に相当するものである)。

【0249】ステップS700として、対象フレーム番号および指定された点の座標を取得する。すなわち、ポインティングデバイス等により指示された画面上の座標が、映像中の画像のどこに相当するかを計算する。さらに、指示された瞬間に再生を行っていた映像のフレーム番号を取得する。なお、フレーム番号ではなく、タイムスタンプを用いてもよい(以下では、フレーム番号として説明する)。

【0250】ここで、映像に付随している映像中の物体の物体領域データから、上記対象フレーム番号において映像中に存在している物体を選択するものとする。これは、物体領域データにおける先頭フレーム番号および末尾フレーム番号を参照することにより容易に実行でき

る。

【0251】当該フレームに（指示できる）物体が存在しない場合には、この時点で処理は終了となる。

【0252】なお、以下の処理は、対象となる物体が複数存在する場合には、各物体について、同時にまたは例えば指示された物体が得られるか最後の物体になるまで順次に繰り返し行うものとする。

【0253】ステップS701では、対象物体について、近似変換パラメータ軌跡から変換パラメータを算出する。この変換パラメータは、参照物体領域から対象物体領域への変換を示している。

【0254】たとえば、図29の例でユーザが画像をクリックした際の対象フレーム番号は10であったとする。この時、動画像が10フレーム/秒であったとすると、タイムスタンプは1.0(秒)である。0003の節点時刻を見ると、時刻1.0は1番目の0.0と2番目の2.0の間であるので、時刻1.0に対応する近似関数は1番目と2番目の節点間の関数ということがわかる。両節点におけるパラメータの値はそれぞれ0.0と-16.8であることが0003の記述からわかり、さらにこの区間の関数IDは1、関数パラメータは4.2\*

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} a & b & c \\ d & e & f \\ 0 & 0 & 1 \end{bmatrix}^{-1} \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix}$$

【0259】点2203の座標を(x'、y')、点204の座標を(x, y)とする。

【0260】ステップS703では、その点が参照物体領域の内部にあるか外部にあるかを判定する。

【0261】内外判定の方法は、使用されている物体領域の表現方法によって異なる。以下では、ピットマップ表現と多角形表現の場合を例として内外判定の方法を説明する。

【0262】参照物体領域がピットマップで表現されているとき、図43の手順に従って、内外判定を行う。参照物体領域のピットマップ表現とは、例えば、物体領域内を1、物体領域外を0で表現することがあげられる。ステップS802によって算出された点の位置のピットマップ情報を、ステップS803で参照して値が1のとき物体領域内にあり、0のとき物体領域外にあると判定できる。

【0263】また、参照物体領域が多角形で表現されているとき、図44の手順に従って、内外判定を行う。例えば、図45に示す参照物体領域多角形において、点2501と多角形の2つの隣り合った頂点を結ぶベクトル2502および2503を生成する。ベクトルは3次元とし、Z座標を0としておく。また、ベクトルの長さが1となるように正規化しておく。2つのベクトルの外積およびベクトル同士のなす角を、1つずつ頂点をずらし

\*といふこともわかる。これらの情報と図28の式を用いてパラメータ $a_0$ の時刻0.0から2.0までの区間の近似関数は $a_0 = 2.1t^2 - 12.6t$ であることが算出される。この式に、 $t = 1.0$ を代入して $a_0$ の値は-10.5と算出される。 $a_1$ のパラメータ値も同様に $a_1 = 5.4t$ に、 $t = 1.0$ を代入して5.4と算出される。

【0255】ステップS702では、指定された点を逆変換して、参照フレームにおける位置を求める。

【0256】先ほどの例では、平行移動モデルでパラメータが $(a_0, a_1) = (-10.5, 5.4)$ と算出されたが、例えばユーザにより指定された座標が(50, 70)であったとすると、平行移動の逆変換は $(50, 70) - (-10.5, 5.4) = (60.5, 64.6)$ となる。

【0257】別の例として、例えば、変換がaffine変換である場合に、図41において点2203から点2204を求めるには、式(8)を用いればよい。

【0258】

【数2】

…(8)

ながら全ての頂点について算出する。外積ベクトルは多角形に対して垂直であり、Z座標は-1か1の値を取る。算出した角度に外積ベクトルのZ座標をかけあわせ加算していくと、点が多角形内部にある場合は-360度もしくは360度になり、多角形外部にある場合はそれ以外の値となる。以上のようにして内外判定を行うことができる。

【0264】また、参照物体領域が順序のついた代表点で表現されている場合に、それをいったんピットマップに展開してから、内外判定を行う方法も可能である。

【0265】まず、参照物体領域の特徴点をその順序に従って結び多角形を生成する。参照物体領域が多角形表現されている場合に内外判定を行う方法については、上述の通りである。生成した多角形をピットマップに展開するには、多角形の全ての辺をあらわす直線の式を求める。それらの式から、多角形内部を表現する連立不等式をたてる。

【0266】簡単な例として多角形が長方形である場合を図46に示す。長方形の各辺の式が図46に示すように得られているとき、多角形内部を表現する連立不等式は式(9)のように求められる。

【0267】

【数3】

$$\left\{ \begin{array}{l} x \geq 2 \\ x \leq 7 \\ y \geq 1 \\ y \leq 4 \end{array} \right.$$

39

【0268】ビットマップの各画素の座標がこの連立不等式を満たせば、その画素の値を1とし、そうでない場合は画素値を0とすることによって、ビットマップへ展開することが可能である。展開されたビットマップを用いて内外判定を行う方法については上述の通りである。

【0269】さて、以上のような判定の結果、ある物体領域が指示されたことが検出された場合には、それを契機として所定の処理が行われる。

【0270】例えば、物体領域データに含まれている関連情報へのポインタを参照し、このポインタ情報を基づいて関連情報を取得し、その表示等を行う（図40の構成例では、これを関連情報呈示部303により行う）。また、関連情報としてプログラムが指定されている場合には、指定プログラムを実行したり、またその他、指定されている所定の動作を行う。なお、物体領域データに関連情報そのものが記述されている場合にはこれを表示等すればよい。

【0271】図47に、関連情報として映像中の物体の説明が付与されている場合の一例を示す。映像800の再生中にポインティングデバイス802により指示された座標が物体801（を近似した図形）の領域内部であった場合、関連情報803が表示される。

【0272】なお、以上の各機能は、ソフトウェアとしても実現可能である。

【0273】また、本実施形態は、コンピュータに所定の手段を実行させるための（あるいはコンピュータを所定の手段として機能させるための、あるいはコンピュータに所定の機能を実現させるための）プログラムを記録したコンピュータ読み取り可能な記録媒体としても実施することもできる。

【0274】本発明は、上述した実施形態に限定されるものではなく、その技術的範囲において種々変形して実施することができる。例えば、各実施形態は単独で説明したが、複数の実施形態を適宜組み合わせてもよい。本願発明は上記各実施形態に限定されるものではなく、実施段階ではその趣旨を逸脱しない範囲で種々に変形することができる。また、各実施形態は可能な限り適宜組み合わせて実施してもよく、その場合組合せた効果が得られる。さらに、上記実施形態には種々の段階の発明が含まれており、開示される複数の構成要件における適宜な組合せにより種々の発明が抽出され得る。例えば、実施形態に示される全構成要件から幾つかの構成要件が削除されても、発明が解決しようとする課題の欄で述べた課題の少なくとも1つが解決でき、発明の効果の

40

…(9)

欄で述べられている効果の少なくとも1つが得られる場合には、この構成要件が削除された構成が発明として抽出され得る。

【0275】

【発明の効果】本発明によれば、映像中における対象となる物体の領域を、参照物体領域と変換パラメータに基づいて記述することにより、映像中の所望の物体の領域を少ないデータ量で記述でき且つその作成やそのデータの扱いも容易にことができる。

【0276】また、本発明によれば、ユーザによる映像中の物体の指示の判定を容易にすることができる。

【図面の簡単な説明】

【図1】本発明の実施形態に係る物体領域情報生成装置の構成例を示す図。

【図2】同実施形態における処理手順の一例を示すフローチャート。

【図3】物体領域のオプティカルフロー算出方法の一例を説明するための図。

【図4】物体領域のオプティカルフロー算出方法の他の例を説明するための図。

【図5】1つの変換パラメータを時間関数で近似表現する例を説明するための図。

【図6】参照物体領域がビットマップ表現されている場合の物体領域データの記述フォーマットの一例を示す図。

【図7】同実施形態における処理手順の他の例を示すフローチャート。

【図8】物体領域情報生成装置の他の構成例を示す図。

【図9】同実施形態における処理手順のさらに他の例を示すフローチャート。

【図10】物体の領域を楕円で近似する例について説明するための図。

【図11】物体の領域を近似する楕円の代表点を求める例について説明するための図。

【図12】近似图形を平行四辺形とした場合の代表点の例について説明するための図。

【図13】近似图形を多角形としたときの代表点の例について説明するための図。

【図14】物体領域近似图形と代表点の例を示す図。

【図15】穴のある物体領域を近似图形で表した例を示す図。

【図16】物体領域の近似图形の代表点同士を対応付けする方法について説明するための図。

【図17】近似に用いる图形の種類と変換パラメータを

41

求めることができる変換モデルとの関係を示す図。

【図18】参照物体領域が図形近似されている場合の物体領域データの記述フォーマットの一例を示す図。

【図19】物体を矩形で近似した様子を示す図。

【図20】傾斜した細長い物体を傾斜を持たない矩形で近似した様子を示す図。

【図21】物体をその傾斜に応じた傾斜を持つ矩形で近似した様子を示す図。

【図22】近似矩形を求める処理手順の一例を示すフローチャート。

【図23】近似矩形から近似楕円を求める方法について説明するための図。

【図24】近似矩形から近似楕円を求める処理手順の一例を示すフローチャート。

【図25】物体領域データ中のパラメータ情報のデータ構造の一例を示す図。

【図26】近似関数を用いた場合のパラメータ情報のデータ構造の一例を示す図。

【図27】物体領域記述情報のデータ構造の他の例を示す図。

【図28】関数IDと関数形式と関数パラメータと制約条件との対応の一例を示す図。

【図29】本発明による代表点軌跡データのデータ構造の特徴を示す図。

【図30】代表点の時系列座標を曲線で近似する処理を示すフローチャート。

【図31】サンプリング情報を含む物体領域データの記述フォーマットの一例を示す図。

【図32】誤差最小となる参照物体領域を選択する処理手順の一例を示すフローチャート。

【図33】物体領域存在区間の中央フレームを参照物体領域とする処理手順の一例を説明するための図。

【図34】参照物体領域区間の先頭を参照物体領域とする処理手順の一例を示すフローチャート。

【図35】参照物体領域区間の先頭を参照物体領域とする処理手順の一例を説明するための図。

【図36】参照物体領域区間の中間を参照物体領域とす\*

42

\*る処理手順の一例を示すフローチャート。

【図37】参照物体領域区間の中間を参照物体領域とする処理手順の一例を説明するための図。

【図38】オプティカルフローを用いて1つの物体を動きが似ている領域に分割する様子を説明するための図。

【図39】1つの物体を複数の領域で記述するための物体領域データの記述フォーマットの一例を示す図。

【図40】同実施形態に係る情報処理装置の構成例を示す図。

10 【図41】指定された位置情報を持つ点が時空間物体領域の内部にあるか外部にあるかを判定する方法について説明するための図。

【図42】指定された位置情報を持つ点が時空間物体領域の内部にあるか外部にあるかを判定するための処理手順の一例を示すフローチャート。

【図43】参照物体領域がビットマップ表現されているときの内外判定方法の一例を示すフローチャート。

【図44】参照物体領域がポリゴン表現されているときの内外判定方法の一例を示すフローチャート。

20 【図45】ポリゴンと点の内外判定を行う方法について説明するための図。

【図46】ポリゴンで表現されている参照物体領域をビットマップ表現に展開する方法について説明するための図。

【図47】物体領域データを利用したハイパーテディアコンテンツの表示例を示す図。

#### 【符号の説明】

2 … 映像データ記憶部

4 … 物体領域処理部

30 5 … 図形近似部

6 … 変換パラメータ算出部

8 … 関数近似部

10 … 領域データ記憶部

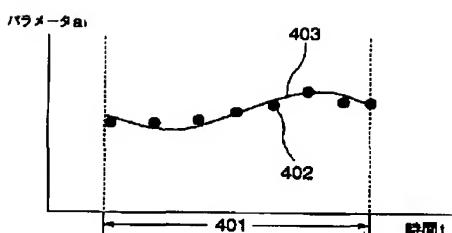
301 … 映像データ表示部

302 … 制御部

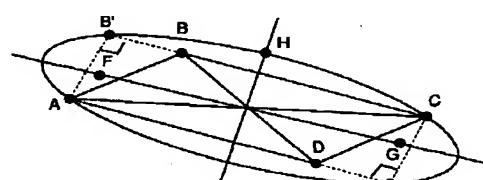
303 … 関連情報表示部

304 … 指示入力部

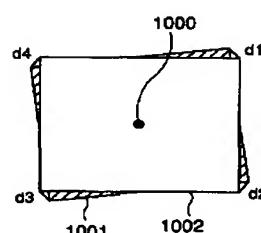
【図5】



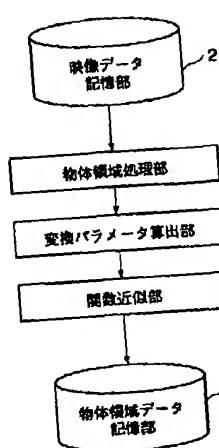
【図10】



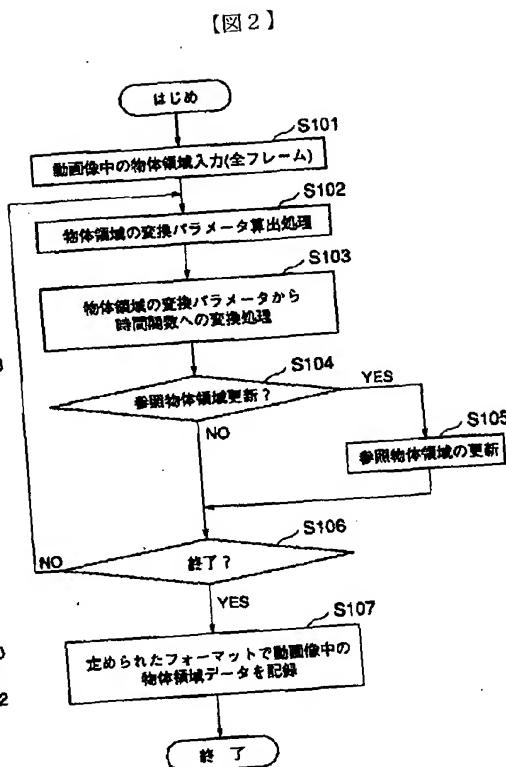
【図16】



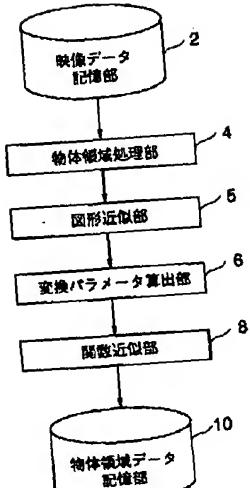
【図1】



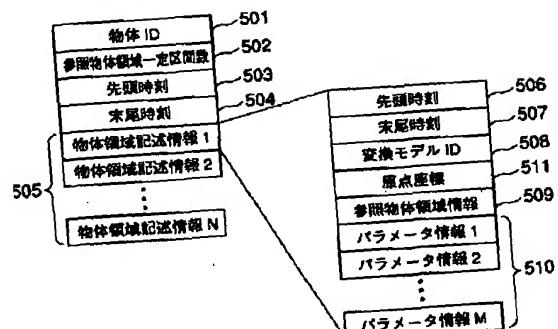
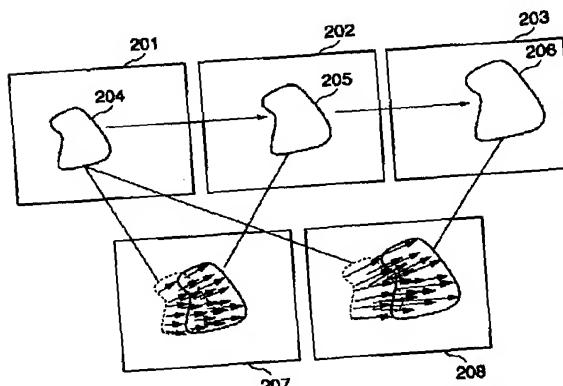
【図2】



【図8】

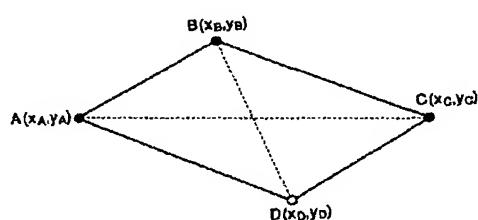
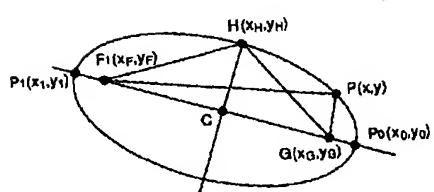


【図3】

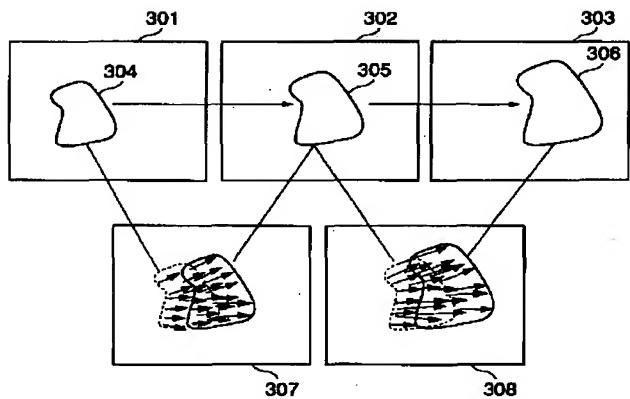


【図12】

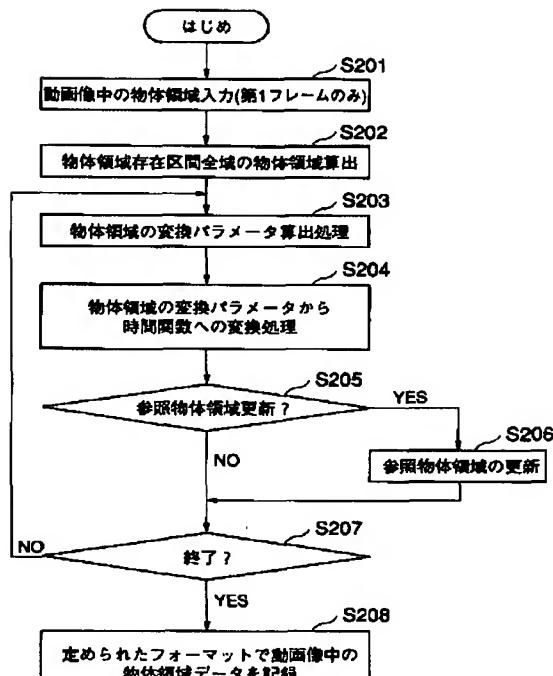
【図11】



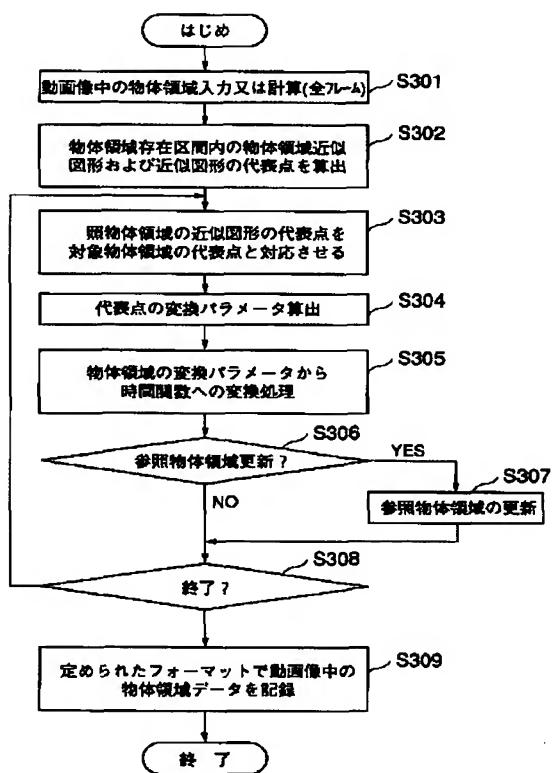
【図4】



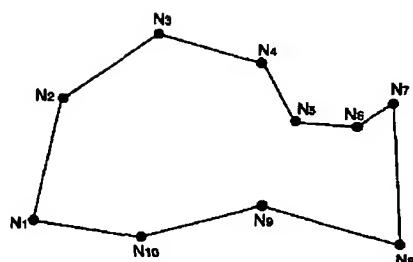
【図7】



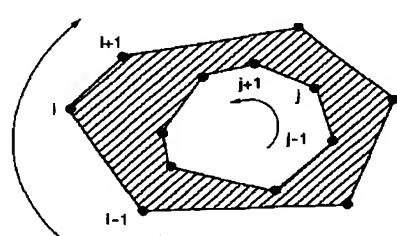
【図9】



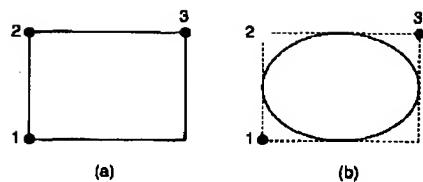
【図13】



【図15】



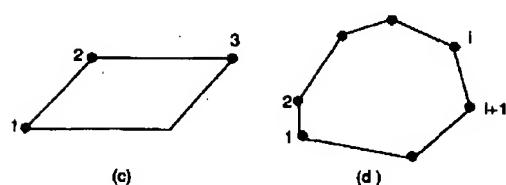
【図14】



【図17】

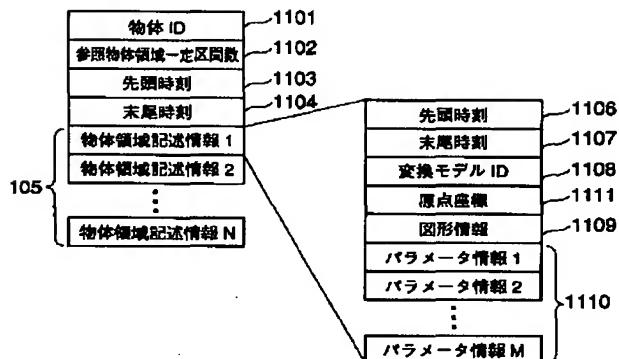
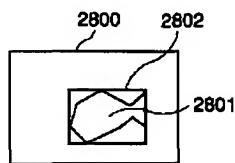
動きモデル 近似图形	縮小	回転	平行 移動	4-parameter 変換	affine 変換	射影 変換	parabolic 変換
矩形・椭円 (傾きなし)	○	×	○	×	×	×	×
矩形・椭円 (傾きあり)	○	○	○	○	×	×	×
平行四辺形	○	○	○	○	○	×	×
多角形 (頂点数>11)	○	○	○	○	○	○	○

【図38】

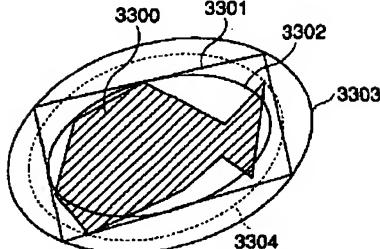


【図18】

【図19】

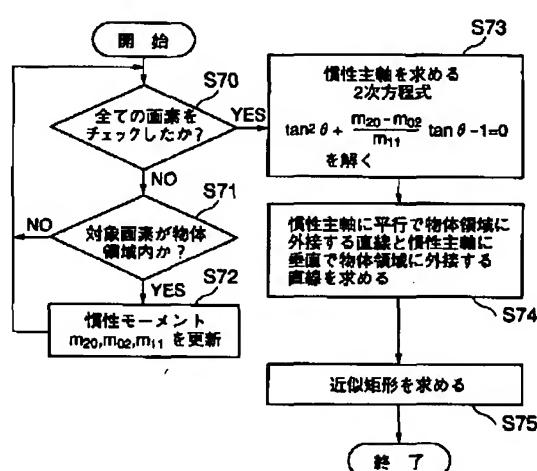
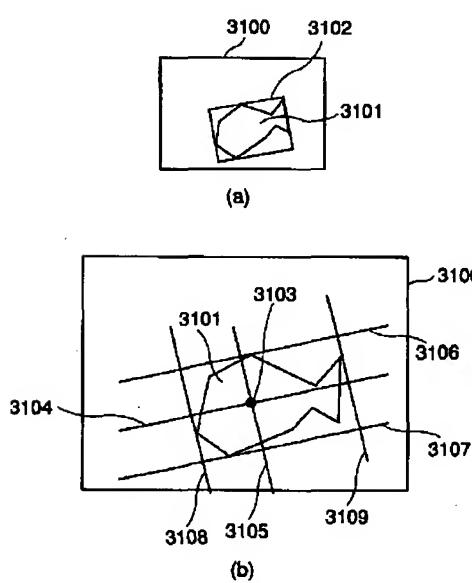


【図23】

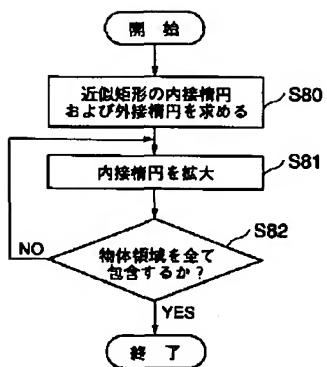


【図21】

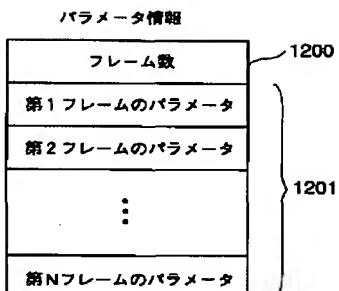
【図22】



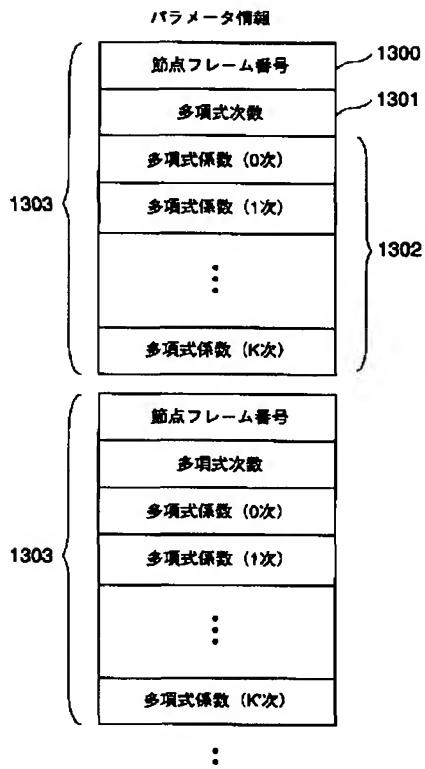
【图 2-4】



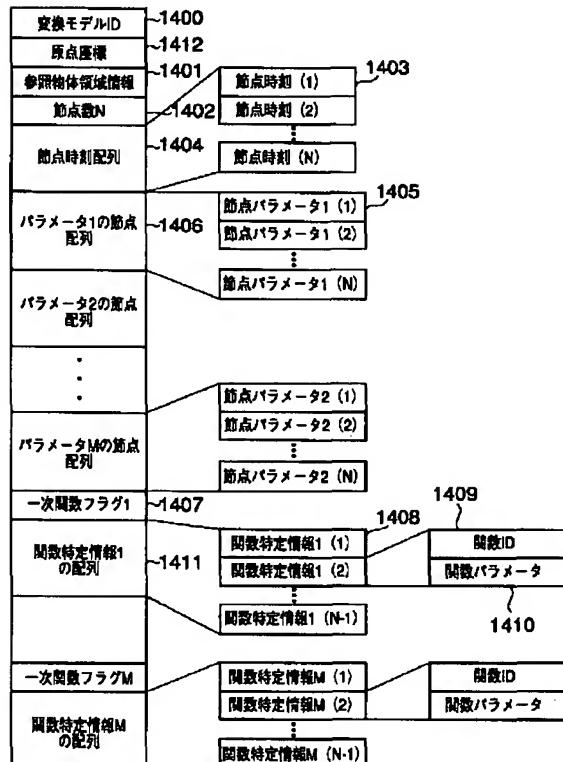
【図25】



【図26】



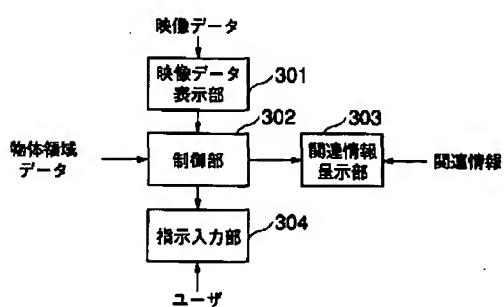
【図27】



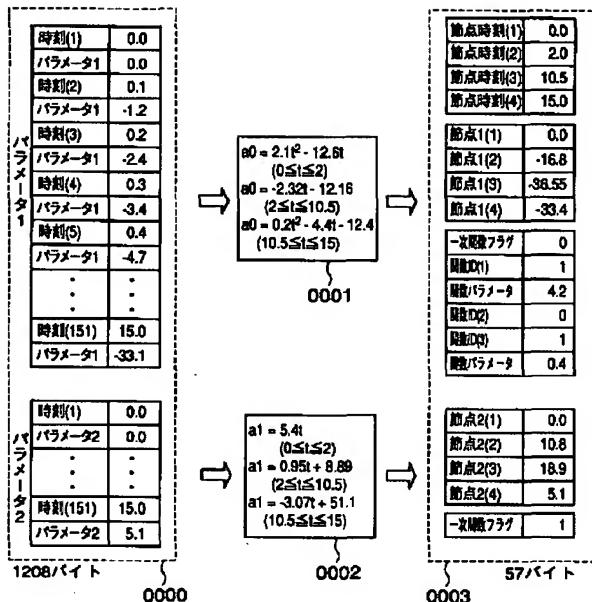
【図28】

関数形式	関数パラメータ	制約条件
なし(関数は定義されない)	なし	なし
$f(t) = f_a + v a(t - t_a)$	なし	$v a = \frac{f_b - f_a}{t_b - t_a}$
$f(t) = f_a + v a(t - t_a) + \frac{1}{2} \alpha a(t - t_a)^2$	$f_a$	$v a = \frac{f_b - f_a}{t_b - t_a} - \frac{1}{2} \alpha a(t_b - t_a)$

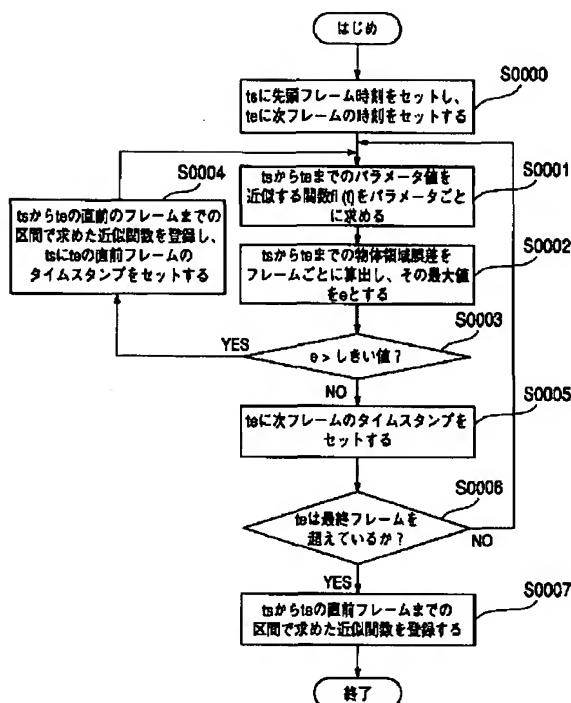
【図40】



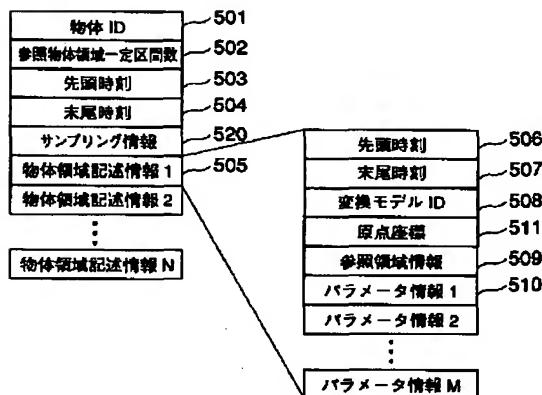
【図29】



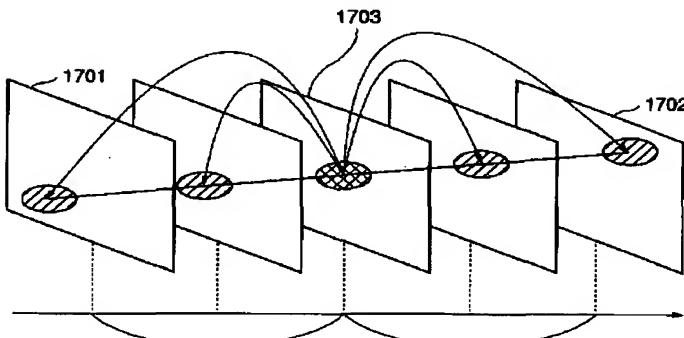
【図30】



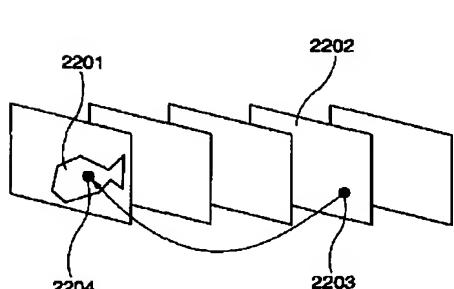
【図31】



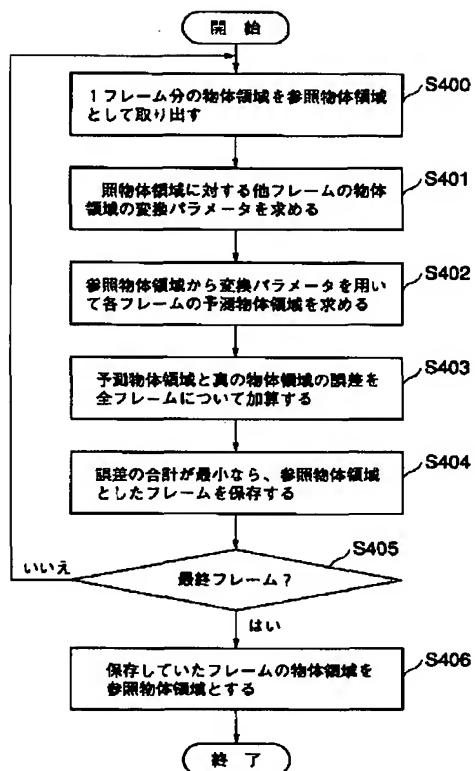
【図33】



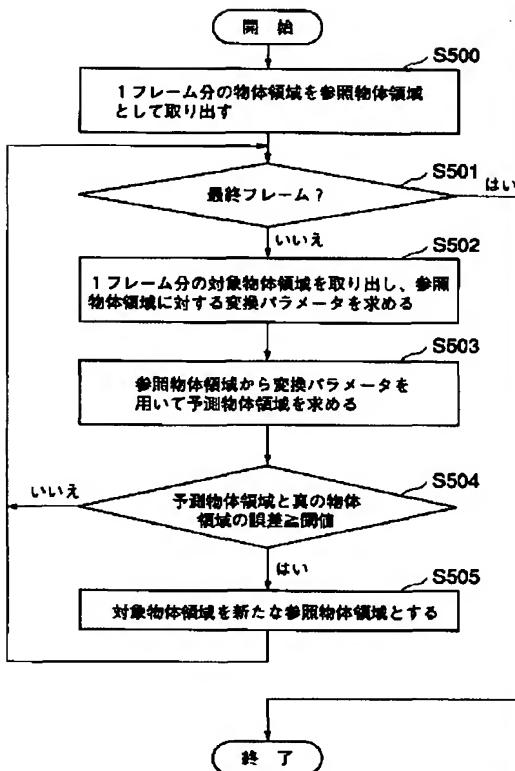
【図41】



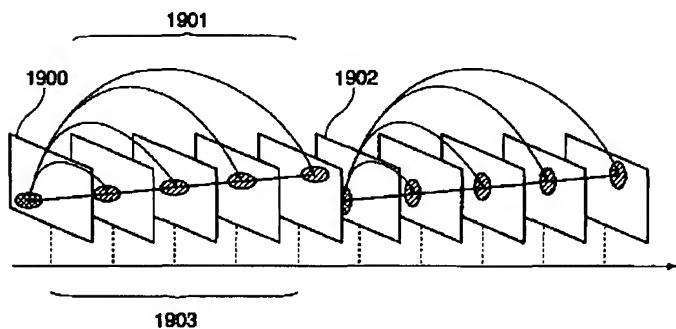
【図32】



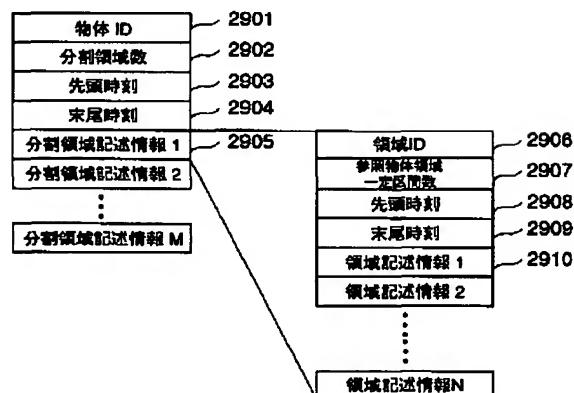
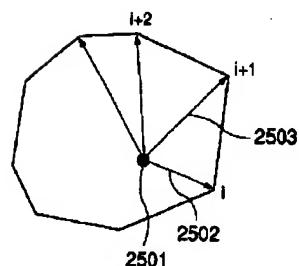
【図34】



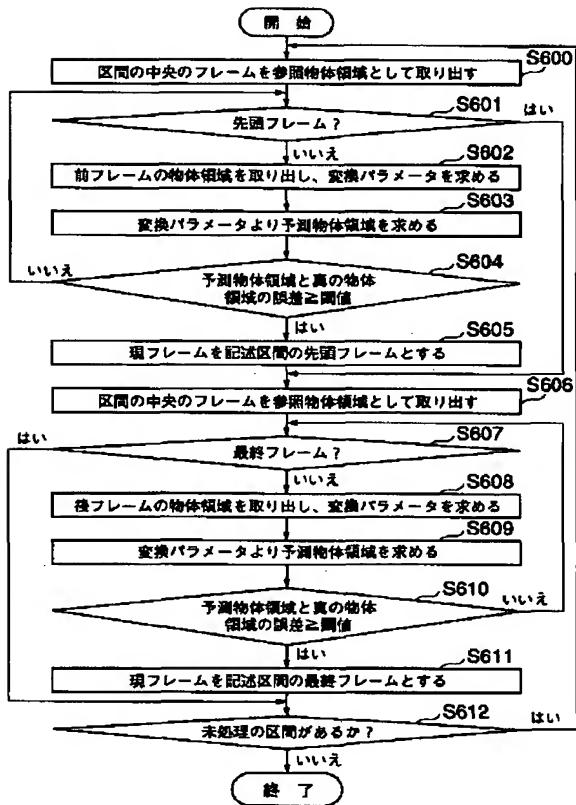
【図35】



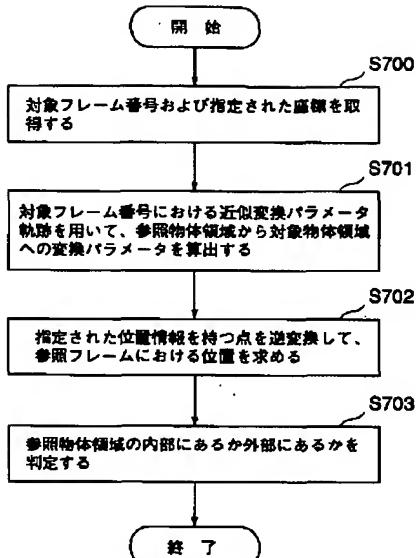
【図45】



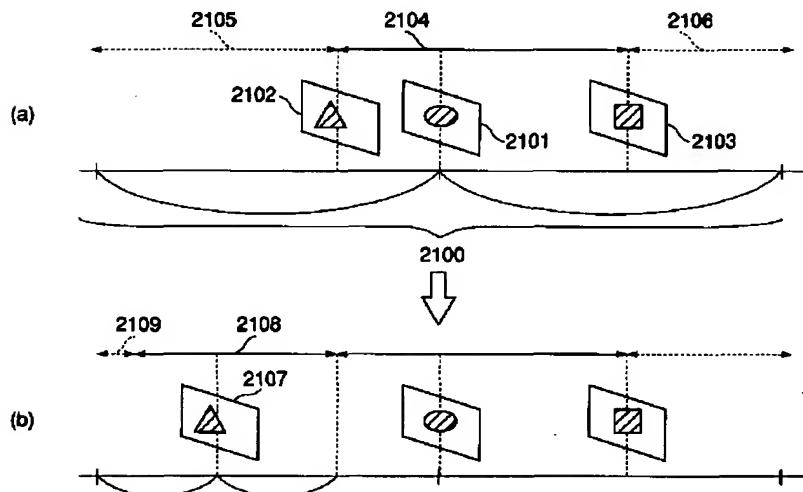
【図36】



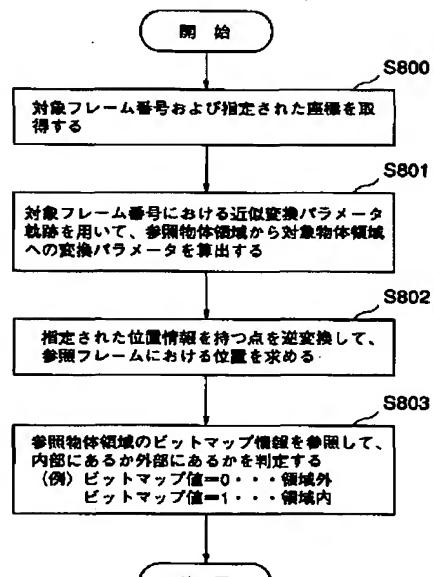
【図42】



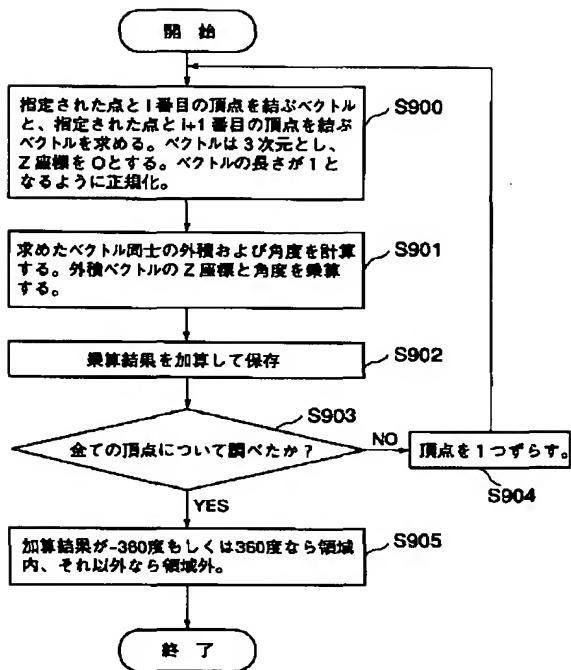
【図37】



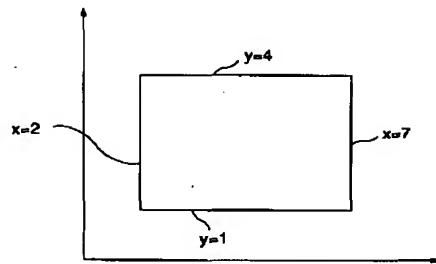
【図43】



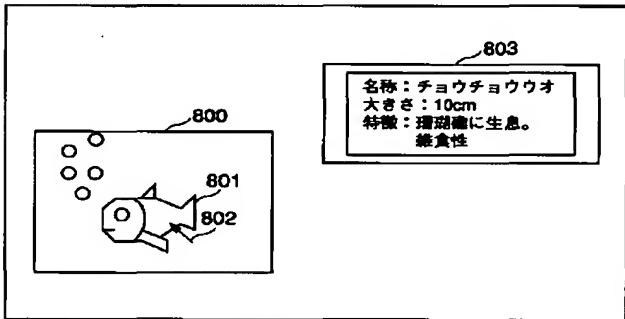
【図44】



【図46】



【図47】



フロントページの続き

(51) Int. Cl. 7

識別記号

F I

データコード (参考)

(72) 発明者 三田 雄志  
 神奈川県川崎市幸区小向東芝町1番地 株  
 式会社東芝研究開発センター内

(72) 発明者 山本 晃司  
 神奈川県川崎市幸区小向東芝町1番地 株  
 式会社東芝研究開発センター内